US DOE SC
Office of Advanced Scientific Computing Research FY08 Joule Software Metric
SC GG 3.1/2.5.2 Improve Computational Science Capabilities

October 10, 2008

Contents

Credits

Application Credits
`DCA++` : Thomas Schulthess (ORNL)
`PFLOTRAN` : Peter Lichtner (LANL)
`GYRO` : Jeff Candy (GA)

Technical Team : (DCA++) Markus Eisenbach, Thomas Maier, Gonzalo Alvarez, Mike Summers (ORNL) , (PFLOTRAN) Richard Mills (ORNL), Glen Hammond (PNNL), (GYRO) Mark Fahey (ORNL)

Additional Credits
Kenneth Roche (ORNL), Ricky Kendall (NCCS ORNL), Doug Kothe (NCCS ORNL)

DOE Program Contacts
Christine Chalk, christine.chalk@science.doe.gov
Daniel Hitchcock, daniel.hitchcock@science.doe.gov
Frederick Johnson, fjohnson@ascr.doe.gov
Michael Strayer, michael.strayer@science.doe.gov

Contacts
Doug Kothe (kothe@ornl.gov)
Kenneth Roche (rochekj@ornl.gov)

The Joule Metric

Public Authorizations
PL 95-91, "Department of Energy Organization Act"
PL 103-62, "Government Performance and Results Act"

0.1 Metric Statement

0.1.1 Joule Metrics

The U.S. Office of Management and Budget (OMB)[1] oversees the preparation and administration of the President's budget, evaluates the effectiveness of agency programs, policies and procedures, assesses competing funding demands across agencies, and sets the funding priorities for the federal government.
OMB has the power of audit and exercises this right annually for each federal agency. According to the *Government Performance and Results Act of 1993 (GPRA)*, federal agencies are required to develop three planning and performance documents:

1. Strategic Plan: broad, three year outlook
2. Annual Performance Plan that is incorporated into the annual budget request: focused, one year outlook of annual goals and objectives; "what results can the agency produce for the taxpayers money?"
3. Performance and Accountability Report: an annual report about the past fiscal year performance; "what results did the agency produce for the taxpayer's money?"

OMB uses its *Performance Assessment Rating Tool (PART)* to perform evaluations. PART has seven worksheets for seven types of agency functions. The function of Research and Development (R&D) programs is included. R&D programs are assessed upon the following criteria:

- does the R&D program perform a clear role?
- has the program set valid long term and annual goals?

[1] http://www.whitehouse.gov/omb

- is the program well managed?
- is the program achieving the results set forth in its GPRA documents?

In FY2003, the Department of Energy Office of Science (DOE SC) worked directly with OMB to come to a concensus on an appropriate set of performance measures consistent with PART requirements. The scientific performance expectations of these requirements reach the scope of work conducted at the national laboratories. The *Joule* system emerged from this interaction. Joule enables the chief financial officer and senior DOE management to track annual performance on a quarterly basis. Joule scores are reported as "success, goal met" (*green light* in PART), "mixed results, goal partially met" (*yellow light* in PART), and "unsatisfactory, goal not met" (*red light* in PART). Joule links the DOE strategic plan[2] to the underlying base program targets.

[2]http://www.er.doe.gov/about/Mission_Strategic.htm

0.1.2 OASCR's FY08 Joule Goals

The Office of Advanced Scientific Computing Research (OASCR)[3] has the following two annual measures that it tracks quarterly:

1. **(SC GG 3.1/2.5.1)** Focus usage of the primary supercomputer at the National Energy Research Scientific Computing Center (NERSC) on capability computing. Percentage of the computing time used that is accounted for by computations that require at least 1/8 of the total resource. *FY08: time used is at least 40%*

2. **(SC GG 3.1/2.5.2)** Improve Computational Science Capabilities: Average annual percentage increase in the computational effectiveness (either by simulating the same problem in less time or simulating a larger problem in the same time) of a subset of application codes. *FY08: efficiency measure is* $\geq$ *100%*

Asserting compliance with these metrics is a critical hurdle each fiscal year for the success of DOE's open science computing effort. This document presents the results of the effectiveness of the computational science capability.

[3] http://www.sc.doe.gov/ascr/About/about.html

0.1.3 Quarterly Tasks Related to SC GG 3.1/2.5.2

This is a year long effort requiring quarterly updates. The general outline of tasks for exercising the software metric are presented by fiscal quarter here.

Q1 Tasks (`deadline: Decemeber 31`)
Identify a subset of candidate applications to be investigated on DOE SC supercomputers. Management (DOE SC and laboratory) decides a short list of applications and computing platforms to be exercised. The Advanced Scientific Computing Advisory Committee (ASCAC) approves or rejects the list. The Q1 milestone is satisfied when a short list of applications and machines is approved.

Q2 Tasks (`deadline: March 31`)
Problems to study on the target machines are determined. The science capability and computational performance of the implementation are benchmarked on the target machines for the defined problems, problem instances. The Q2 milestone is satisfied when benchmark data is collected and explained. In the case that an application is aiming to achieve a new result, the Q2 milestone is satisfied by providing a detailed discussion of current capability, a discussion of why the capability is insufficient, and a description of the new capability being developed.

Q3 Tasks (`deadline: June 30`)
The application software is enhanced for efficiency, scalability, science capability, etc. The Q3 milestone is satisfied when the status of each application is reported at the Q3 deadline. Corrections to Q2 problem statements are submitted during this quarter.

Q4 Tasks (`deadline: September 30`)
Enhancements to the application software continue as in Q3. The enhancements are stated and demonstrated on the machines used to generate the baseline information. A comparative analysis of the Q2 and Q4 data is summarized and reported. The Q4 milestone is satisfied by asserting that the enhancements made to the application software are in accordance with the efficiency measure and type of enhancement -efficiency, scalability, or new result.

0.2 FY08 Results

(SC GG 3.1/2.5.2) Improve Computational Science Capabilities: Average annual percentage increase in the computational effectiveness (either by simulating the same problem in less time or simulating a larger problem in the same time) of a subset of application codes. *FY08: efficiency measure is* $\geq 100\%$

Each application is discussed and the problems benchmarked are described in the respective application sections -the long write-ups. A brief description of the machine used for the benchmarks is given. A summary of measured results is followed by subsequent subsections on results and analysis of the measured results per application, and a section on tools and analysis to orient the reader.

0.2.1 FY08 Target Machine: `jaguar.ccs.ornl.gov`

The Cray XT4 cluster, *Jaguar*, at Oak Ridge National Laboratory's (ORNL) National Center for Computational Sciences (NCCS) was used to exercise OASCR's FY08 Joule software metric.

Jaguar has a total of 7,832 XT4 compute nodes. The compute nodes are operated by the Compute Node Linux (CNL) software -a variant of Linux. Each compute node is a single-socket, quad-core (Budapest) 2.1 GHz AMD Opteron (TM) processor ($75Watt$) with 8 GB of unbuffered memory, 2 MB of shared L3 cache, 512 KB L2 cache per core, and 64 KB instruction and 64 KB data L1 caches per core. Of the 7,832 sockets, 932 have 667 MHz DIMMs and 6900 have 800 MHz DIMMs. In the best case, local memory bandwidth is $\sim 12.8GB/second$ per node.

Jaguar has 112 input / output (i/o) and login / service nodes. Each of these is a 2.6 GHz dual-core AMD Opteron (TM) chip with 8 GB of memory per node. The i/o and service nodes are running a variant of SuSE Linux. Approximately 600 TB are available in the scratch filesystems and support massive i/o parallelism through the Lustre filesystem software.

All nodes are linked by HyperTransport (HT) to Cray's proprietary SeaStar2+ chips which are used to construct a 3d-torus communication network between nodes. There are 6 switch ports per Cray SeaStar2+ chip and each port has bandwidth $9.6GB/s$. The best case bandwidth between the compute node and the SeaStar2+ interconnect chip is $6.4GB/s$. Thus, the injection bandwidth is half this, or $3.2GB/s$.

For further information, the NCCS website[4] describes the system and its software stack and is sufficiently detailed for the purposes of this report. For information on the Cray XT4 platform see
http://www.cray.com/Assets/PDF/products/xt/CrayXT4Blade.pdf. For chip specific information on the single socket 1000 series see
http://www.amd.com/us-en/Processors/ProductInformation/0,,30_118_8796_15226,00.html.

0.2.2 Summary of Results

The measured results are interpreted against a weak scaling model with the aim to satisfy the language of the metric, "*simulating a larger problem in the same time.*" It is not generally the case that the applications are defined by a single parameter or that the physical scaling of the problem parameter(s) will yield a change in computational complexity linearly related to the computational complexity of the original problem (ideal weak scaling). However, usually an analysis can be made that projects the linear increase in computational complexity for the application to an affiliated change in problem specific parameters.[5] The essential feature of scaling various parameters of a

[4] http://www.nccs.gov/computing-resources/jaguar/
[5] See section[0.2.6] for a detailed example.

problem is to focus the analysis or improve the resolution of the computed results.

Technically, the program binary is the algorithm for the problem on the target machine and the computational complexity of each problem instance can be deduced directly by monitoring the values of the various program counters for the various functional units activated during program execution. In other words, the complexity of the problem is defined by the required resources and the work conducted to actually execute it. This measure of work is fairly basic from the hardware perspective and can be derived from system observables such as number of processes PEs (1-1 with cpu-core count in this study) dedicated to executing the program, execution time $TIME$, total number of instructions INS executed[6], magnitude of the memory demand in $BYTES$, etc.

Overview. Before discussing the individual application benchmarks, the aggregated machine event information collected while executing the Q2 and Q4 benchmarks is presented in tables[1,2] and compared in table[3]. This approximates the total computational complexity to execute the original and scaled set of FY08 benchmarks -all on Jaguar. The interpretation is that the aggregated data reveals *total satisfaction with the weak scaling metric from the hardware event perspective*. That is, the scaled FY08 problem set computed 3.7623 times the number of instructions (3.953 times the floating point operations) with 3.8869 times the number of processes in .9998 percent of the time it took to compute the original problem set. So, the Q4 problem set is $\sim 4X$ *larger* than the Q2 problem set and executes on $\sim 4X$ more processes than in Q2 and terminates in the same (actually less) time as the Q2 problem set. The impression is that the applications scale linearly on Jaguar and this is essentially true for the carefully crafted studies made in FY08. The insight gained owing to the increased information is described as a highlight in this summary, whereas the context and significance are devloped in the detailed descriptions of each application respectively.

If it is confusing, please note that some of the applications were also improved for efficiency or simply performed better from the machine perspective when executing a larger problem.

Application	**DCA++**	**GYRO**	**PFLOTRAN**
Metric	time / disorder configuration	timesteps / second / process	time / dof / PE
Problem Q2	$N_{dis} = 64, N_c = 16, N_t = 150$	$\mu = 30$, 20 timesteps	$64.8M$DOFs, 201 flow, transport steps
Hardware Used Q2	7808 PEs	4608 PEs	4000 PEs
Walltime Q2	25339 s	69.01 s	2594 s
Instructions Q2	5.1805e17	8.8854e14	2.2222e16
FLOPs Q2	4.6270e17	2.8701e14	1.2898e15

Table 1: Q2 FY08 Joule Software Summary of Benchmark Data.

0.2.3 DCA++ analysis

The two-dimensional (2D) Hubbard model is explored with the Dynamical Cluster Approximation (DCA) in conjunction with the Hirsch-Fye QMC (HF-QMC) algorithm. The approach in these simulations is to solve the quantum many-body problem at the atomic and nano-scale exactly with QMC, and to account for the macroscopic length scales within a mean-field approximation by self-consistently and coherently embed-

[6]The instruction set is not to be confused with basic operations that are defined in the language of the instruction set of the chip. For instance, in a single cycle, a single cpu-core (1 PE) on Jaguar can compute 4 double precision mathematical operations (fused multiply and add).

Application	**DCA++**	**GYRO**	**PFLOTRAN**
Metric	time / disorder configuration	timesteps / second / process	time / dof / PE
Problem Q4	$N_{dis} = 256, N_c = 16, N_t = 150,$	$\mu = 40$, 20 timesteps	$129,635,520$ DOFs , Q2 stepping
Hardware Used Q4	31232 PEs	24576 PEs	8000 PEs
Walltime Q4	23791	1248.58 s	2958.36 s
Instructions Q4	1.93e18	5.5680e16	5.0374e16
FLOPs Q4	1.8126e18	1.9856e16	2.8603e15

Table 2: Q4 FY08 Joule Software Summary of Benchmark Data.

TOTALS	Q2	Q4	ratio (Q4 : Q2)
$\sum Walltime$	28002.01 s	27997.94 s	.9998
$\sum PEs$	16416	63808	3.8869
$\sum Instructions$	5.4116e17	2.036e18	3.7623
$\sum FLOPs$	4.6427e17	1.8353e18	3.953

Table 3: FY08 Joule software summary of aggregated Q2 and Q4 benchmark data.

ding the cluster solution into an effective medium. In the limit of infinitely large clusters, the DCA/QMC recovers the exact solution.

One must investigate models that include disorder to analyze physical effects contributing to the behavior of high-Tc supercondutors. The total number of disorder configurations N_{dis} is proportional to 2^{N_c}, where N_c is the number of cluster sites.

The benchmarks are intended to show the scaling of the disorder configurations that are accessible during a fixed wall clock time with the number of available compute nodes. The problem is representative of production runs, but the number of Green's function measurements has been reduced to a number that allows the timing runs to finish in approximatly fixed walltime. The ability to weakly scale to larger numbers of disorder representations will enable one to obtain the disorder dependence of the superconducting transition temperature with increasingly better statistics as the available machines grow in size.

The following parameters were used for the Hubbard model parameters in both the Q2 and Q4 DCA++ benchmarks.

- Number of sites: $N_c = 16$
- Number of time slices: $N_t = 150$
- Hubbard U: $U \in \{0.5, 1.5\}$ randomly chosen for a disorder configuration.
- Number of selfconsistent measurements: 20
- number of warm-up steps: 40

The number of disorder configurations was increased from $N_{dis} = 64$ in Q2 to $N_{dis} = 256$ in Q4. This allows the determination the time to execute each disorder configuration t_{dis}.

During the time measurements the calculation of self energies was excluded since this is a fixed amount of computation independent of the number of selfconsistent measurements and a production run will require ≈ 10 times the measurements performed during the benchmark runs.

Performance data for the Q2 and Q4 runs is presented in table[0.2.3] and the ratios of the data in table[0.2.3]. One concludes that the DCA++ disorder study scales almost as the ideal linear case.

	N_D	PEs	INS	FLOP	Walltime
Q2	64	7,808	$5.1805e17$	$4.627e17$	25339 s
Q4	256	31,232	$1.93e18$	$1.8126e18$	23791 s

Table 4: DCA++. Performance results for the Q2 and Q4 disorder configuration scaling benchmarks.

Observable	Ratio(Q4 : Q2)
N_D	4
Walltime	.9389
PEs	4
Instructions	3.725
FLOP	3.91

Table 5: DCA++. Q4 to Q2 benchmark data ratios for the disorder configuration scaling exercises.

The significance of the scalability of the software over the number of disorder configurations is that one can investigate the transition to a superconducting state with d-wave symmetry. The order parameter that signals a transition to this state is given by

$$\Delta_d^\dagger = \sum_{\dot{R}^0} g(\dot{R}^0) c^\dagger_{\dot{R}^0,\uparrow} c^\dagger_{-\dot{R}^0,\downarrow}\,, \tag{1}$$

where $g(\dot{R}^0) = \cos \dot{R}^0_x - \cos \dot{R}^0_y$ is a d-wave form-factor. In linear response theory, the pair-field susceptibility formed from this order parameter

$$P_d = \int_0^\beta d\tau \langle T_\tau \Delta_d(\tau) \Delta_d^\dagger(0)\rangle \tag{2}$$

diverges at the transition temperature T_c to the d-wave superconducting state.

Using the Bethe-Salpeter equation, the pair-field susceptibility may also be written in terms of a particle-particle vertex function Γ^{pp}

$$P_d = P_d^0 + P_d^0 \Gamma^{pp} P_d\,. \tag{3}$$

Here, P_d^0 and Γ^{pp} are matrices of size $NL \times NL$, where N is the number of lattice sites and L the number of Matsubara-frequencies, P_d^0 is the pair-field susceptibility of the non-interacting part of the model and the right hand side contains an implicit sum over the matrix elements. In the DCA, the vertex function Γ^{pp} is approximated by its corresponding cluster quantity which is a smaller matrix of size $N_c L \times N_c L$. This matrix is calculated from the cluster pair-field susceptibility which is measured in the HF-QMC process in the last DCA iteration. Since Eq. (14) can be written as

$$P_d = \frac{P_d^0}{1 - \Gamma^{pp} P_d^0}\,, \tag{4}$$

one can conveniently determine instabilities by calculating the eigenvalues and eigenvectors of the pairing matrix $\Gamma^{pp} P_d^0$, i.e., solving

$$-\frac{T}{N_c} \sum_{K'} \Gamma^{pp}(K, K') P_d^0(K') \phi_\alpha(K') = \lambda_\alpha \phi_\alpha(K)\,. \tag{5}$$

The susceptibility diverges when the leading eigenvalue λ_α crosses one which determines T_c. The symmetry of the ordered state is then given by the K dependence of the corresponding eigenvector $\phi_\alpha(K)$.

At low temperatures the leading eigenvector has $d_{x^2-y^2}$-symmetry. The leading eigenvalue is shown as a function of temperature in Fig. 1

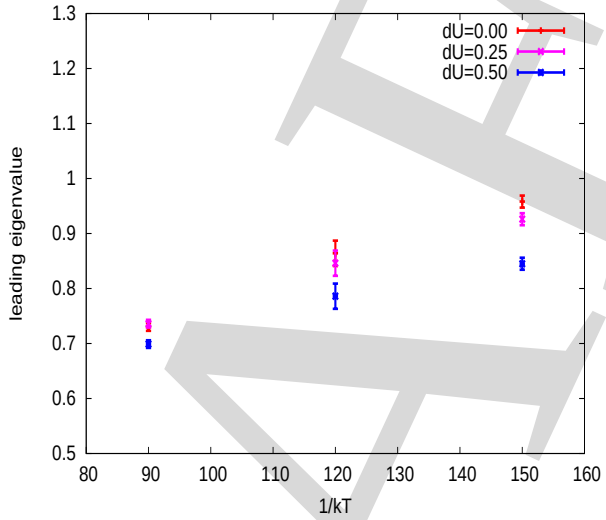

Figure 1: DCA++. The leading eigenvalue λ_d of the Bethe-Salpeter equation, Eq. (16), calculated on a 16-site cluster for $U_i = 4t(1 + \xi_i dU)$, where ξ_i is a random number with value ± 1 for three different values of disorder.

In the system without disorder ($dU = 0$), the leading eigenvalue crosses one at the transition temperature $T_c^{\text{clean}} = 0.08t$. In the disordered system, T_c is reduced, although the reduction is only significant for $dU = 0.5$. One sees that as the disorder strength is increased, T_c decreases. It therefore must be concluded that T_c is suppressed by disorder in the interaction strength.

DCA++ Short Summary. In Q2, 64 disorder configurations were computed over 16 cluster sites with DCA++. The computation utilized 7808 PEs for 25339 seconds. In Q4, 256 = 4 * 64 disorder configurations were computed on 4 * 7808 = 31232 PEs for 23791 seconds. It is observed that as the disorder strength is increased, T_c decreases. It therefore must be concluded that T_c is suppressed by disorder in the interaction strength. This is a significant result.

0.2.4 GYRO analysis

GYRO returned benchmark data in both Q2 and Q4 for 10 and 20 timestep runs so that the cost of their production runs could be deduced by analysis since there is a small (albeit nonlinear function of μ) cost of startup each run prior to entering the time evolution phase that dominates the complexity of a run. Table[6] includes the benchmark information for the 10 and 20 time step runs from both Q2 ($\mu = 30$) and Q4 ($\mu = 40$) benchmarks.

Thus, the normal work signature for 10 timesteps in GYRO can be deduced by taking the differences between observables measured in the 20 and 10 timstep runs. The differences are presented in table[7].

Problem	Hardware	Wall time	Instructions	FLOP
$\mu = 30$, 10 timesteps	4608 PEs	51.78 s	6.6444e14	2.1869e14
$\mu = 30$, 20 timesteps	4608 PEs	69.01 s	8.8854e14	2.8701e14
$\mu = 40$, 10 timesteps	24576 PEs	1095.82 s	4.4377e16	1.3768e16
$\mu = 40$, 20 timesteps	24576 PEs	1248.58 s	5.5680e16	1.9856e16

Table 6: GYRO Benchmark Data for 10 and 20 timestep benchmarks in Q2 and Q4.

	Q2	Q4
Time	17.23 s	152.75 s
Instructions	2.2410e14	1.2202e16
FLOP	6.8320e13	6.0882e15

Table 7: GYRO benchmark data normalized for 10 physical timesteps and removing startup cost in Q2 and Q4.

Finally, we conduct a weak scaling analysis over the physical parameter μ and the performance data captured during the GYRO benchmarks. Table[8]

Observable	Ratio(Q4 : Q2)
Time	8.865
PEs	5.333
Instructions	54.448
FLOP	89.112

Table 8: GYRO benchmark data Q4 to Q2 ratios for the normalized workflow.

It is noted that the physical resolution was improved by increasing the value of μ, the magnetic moment per unit mass, from 30 to 40. GYRO researchers estimate the cost of executing their program at fixed domain size and velocity-space resolution to be related to μ by $cost \sim O(\mu^{3.5})$. Here this ratio is 2.737. The value $\mu = 60$ is the physical target value for a pure deuterium plasma problem.

In the Q4 benchmark, the distribution function was discretized over 3.9 billion gridpoints as compared to 69 million gridpoints in Q2 increasing both the domain and velocity-space (as well as other grid observables) resolution of the simulation necessary to treat the ion-scale modes that dominate the transport physics. The ratio of the number of gridpoints that must be evaluated each time step in the respective benchmarks ($3.9e9/69e6 \sim 56.521$) accounts for the increased complexity reported in table [8].

Compared to the weak scaling factor (as deduced by the increased hardware) of 5.333 it is noted that GYRO exhibits exceptional weak scaling on jaguar. Indeed, note that $\frac{INS/TIME(Q4)}{INS/TIME(Q2)} \sim= 6.142 > 5.333$. (make the same analysis w/ FLOP counts and it is seen that there is an increased floating point work rate in the Q4 run, that is $\frac{FP\_OP/TIME(Q4)}{FP\_OP/TIME(Q2)} \sim= 10.052$)

GYRO Short Summary. The GYRO summary is made for the normalized workload (removing the initialization cost). In Q2, 69 million grid points to resolve the domain

and velocity space were evaluated for 10 time-steps with the magnetic moment per unit mass $\mu = 30$. The computation utilized 4608 PEs for 17.23 seconds. In Q4, 3.9 billion grid points represented the domain and velocity space and with a magnetic moment per unit mass $\mu = 40$. The Q4 benchmark utilized 24576 PEs for 152.75 seconds to compute 10 physical time-steps. The target physical value is $\mu = 60$ to resolve a pure deuterium plasma problem.

0.2.5 PFLOTRAN analysis

From table[9] it is noted that from the machine perspective PFLOTRAN demonstrated weak scaling by increasing the spatial resolution of their problem from $\delta x = \delta y = 2.5m$ to $\delta x = \delta y = 2.5/\sqrt{2}m$. (in each case $\delta z = .1667m$)

This corresponds to a spatial grid of the Hanford 300 area that has $540 \times 1000 \times 120 := 64,800,000 DOFs$ in Q2 and $764 \times 1414 \times 120 := 129,635,520 DOFs$ in Q4. The increase in spatial resolution helps to separate numerical and physical convergence issues the velocity flow fields.

Observable	Ratio(Q4 : Q2)
DOF	2.005
Time	1.14
PEs	2
Instructions	2.2668
FLOP	2.217

Table 9: PFLOTRAN benchmark data Q4 to Q2 ratios.

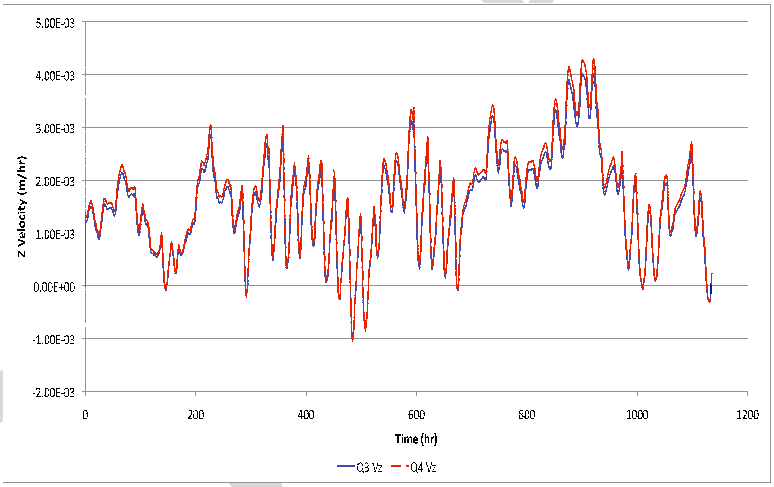

Figure 2: PFLOTRAN. Comparison of the z velocity in Q2 (blue) and Q4 (red) runs.

The fact is that, on a factor of two more processes, the Q4 benchmark computed 2.2668 more instructions in 1.14 more time than in Q2 (e.g. $\frac{2.2156}{1.11} \sim 1.996$ -compared to 2).

By increasing the number of total degrees of freedom, the fidelity of the computed velocity fields is also improved. To test the improvement in the solution, the velocity was compared between runs Q2 and Q4. The x- and y-velocities were very similar indicating convergence. However, slight changes were present in the z-velocity, especially in the peak z velocity. This is shown in Figure 2. These peak velocities have the potential of significantly impacting uranium transport at the Hanford 300 Area, the solutions from which are used to assess risk to the environment (i.e. neighboring Columbia River).

PFLOTRAN Short Summary. In Q2, a 540 x 1000 x 120 spatial grid of the Hanford 300 area was computed utilizing 4000 PEs for seconds. In Q4, the spatial resolution was refined to a 764 x 1414 x 120 spatial grid of the Hanford 300 area was computed utilizing 2 * 4000 = 8000 PEs for seconds. A physically relevant difference in the z-velocities of the contaminant flow fields was observed as a result of the improved resolution -that which may significantly impact the uranium transport in the area and therefore the risk assessment to the environment of these flows.

0.2.6 Tools and How to Use Tools for Scientific Discovery

The acceptability of computed results is defined by the problem. In OASCR's Joule software exercises this FY, the complexity of executing the problems was directly deduced according to machine events measured with supported system software on the target platform. Knowing or having an estimate of the theoretical complexity of the problem is helpful but is often not the case in many of DOE's applications. In any case, the complexity of the problems we can study or solve computing is bound by the hardware and it would be nice to have a method for calibrating the complexity of software instances for certain problems. For instance, one could then order the problems according to their actual computability and match them to target platforms and machine scales.

A detailed example may help here. Suppose our application problem is to have a computer program that executes (on Jaguar) the common math operation $C \leftarrow \alpha AB + \beta C$ where A, B, C are all rank two arrays of double precision, complex numbers with dimensions $A \in [m, n]$, $B \in [n, p]$, $C \in [m, p]$ and α, β are double precision, complex numbers. The problem, $P(m, n, p)$, has complexity that is well described by the storage demands, $O(mn + np + mp + 2)$ *complex numbers*, and floating point operation count, $P(m, n, p) \sim O(8mpn + 13mp)$. This problem's complexity (like all program instances) can be calibrated with machine capabilities (even if we did not have a theoretical estimate) by counting the instructions and specifically floating point instructions completed to execute an instance on Jaguar.

To further simplify the remainder of this discussion, let $m = n = p$. In this case the theoretical complexity of $P(n)$ is $\sim O(3n^2 + 2)$ complex numbers and $\sim O(8n^3 + 13n^2)$ floating point operations. (In the real number case, the problem floating point complexity for $m = n = p$ is $P(n) \sim O(2n^3 + 2n^2)$ and the storage becomes $O(3n^2 + 2)$ real numbers.) Please look at figure[3] since the data presented there is used in this discussion. The figure presents the process count and measured total instructions, total floating point instructions, and execution time (measured with the PAPI software) for a handful of instances of $P(n)$ on Jaguar.

Hardware Events. The first thing to check is the quality of the counts returned by the tool. Consider first, $P(n = 2048)$. The theoretical complexity of this instance is $P(2048) = 6.877400269e10\ FLOP$ for the complex representation and when executed with one process, the tool measured $69084381184\ FLOP$. The difference is less than a half percent. The theoretical complexity in the real case is $P(2048) = 1.718825779e10\ FLOP$ and the measured count was $17251172352\ FLOP$. The difference is less than a half percent. Next, consider $P(8n = 8(2048) = 16384)$, which is an example of increasing the problem parameter n by a constant factor -here a factor of 8. The theoretical complexity for the complex representation is $P(16384) = 3.518786175e13\ FLOP$ and

the measured count was $3.65606E+13FLOP$ for eight process execution. The difference is $\sim 3.75\%$. It is noted that the difference increased as the number of processes increased, an overhead incurred due to concurrency and organizing the computation in the distributed environment. In fact the many-process and single process algorithms for our problem are quite different -the parallel version being more complicated. The important feature for the user is that the parallel version completes execution faster.

Performance. In theory each Jaguar cpu-core can execute $\frac{4FLOP}{cycle}$ at rate $2.1e9Hz$ for a peak performance (ignoring many realistic chip features such as memory capacity, bandwidth, and latency) of $\frac{8,400,000,000FLOP}{second} = 8.4e9\ FLOPs$. Thus, a lowest bound on the time required to execute the complex representation of $P(n = 2048)$ is $\frac{8*2048^3+13*2048^2}{8.4e9}seconds = 8.49seconds$. The example program ran in $9.999seconds$ and we conclude that the program did not execute the floating point operations at the ideal rate -nevertheless, one now has insight as to the efficiency the program executed on the target. ($\frac{8.49}{9.999} \sim .849$) Yes, for this problem the efficiency of productive work is impressive as expected since $O(n^3)$ work is performed on each $O(n^2)$ operands. For the parallel, $8\ PE$, $n = 16384$ case, the lowest bound on the time to execute the complex representation of the problem is taken to be $\frac{8*(8*2048)^3+13*(8*2048)^2}{8\cdot 8.4e9}seconds = 523.628seconds$. The example program ran in $686.7seconds$. ($\frac{523.628}{686.7} \sim .762$ -despite the overheads involved in the parallel algorithm , the performance is quite impressive)

Scalability. The next thing to address is scalability of our program. One way to talk about this is from the problem perspective, in other words consider how execution time and resources change as the parameter that dictates our problem, n, is increased. The figure can again be used. For example, does our problem scale linearly in the weak scaling case? We already know that it does not, but this is also clear from the data for computing the complex version of the problem for n=2048 and n=16384=8*2048 on 1 and 8 processes respectively. If the problem did scale linearly with n, then the compute times would be the same (or within a few percent difference) and they clearly are not. It is noteworthy that the ratio of the complexity of these problem instances is not linear, $\frac{8\cdot 16384^3+13\cdot 16384^2}{8\cdot 2048^3+13\cdot 2048^2} \sim 511.64 \sim 8^3 (= 512) \neq 8$. The cubic term dominates the scaling behavior, as expected. Now, in consideration of the language of the metric, suppose we wanted to execute the larger problem in the same compute time. To obtain a good guess for the ballpark of cpu-cores necessary to satisfy this constraint, solve

$$\frac{8 * 16384^3 + 13 * 16384^2[FLOP]}{\frac{4FLOP}{cycle\cdot core} \cdot 2.1e9[cycles/second] \cdot k[cores]} = TIME_{n=2048} = 9.999seconds$$

for k cores. We find that to satisfy the language of the metric requires $\sim 419[cpu-cores]$ or PEs in theory. For many DOE funded applications, the complexity of the original problem (as opposed to the larger one) is so large to begin with in process count that it is not practical to compute the corresponding larger problem simply due to not having a sufficient increase in computing power (PEs) in hardware to support the demand. In this case, the problem's complexity is calibrated in a linear weak scaling mode.

Finally, consider the context of constraining the multiplication problem to be studied in a linear weak scaling mode. The weak scaling constraint for this problem is

$$\frac{8n_{Q4}^3 + 13n_{Q4}^2[FLOP]}{\frac{4FLOP}{cycle\cdot core} \cdot 2.1e9[Hz] \cdot k_{Q4}[cores]} = \frac{8n_{Q2}^3 + 13n_{Q2}^2[FLOP]}{\frac{4FLOP}{cycle\cdot core} \cdot 2.1e9[Hz] \cdot k_{Q2}[cores]}$$

or

$$\frac{8n_{Q4}^3 + 13n_{Q4}^2}{k_{Q4}}[seconds] = \frac{8n_{Q2}^3 + 13n_{Q2}^2}{k_{Q2}}[seconds]$$

and we want the Q4 problem to be *larger* than the Q2 problem by factor $\gamma \geq 1$. Thus, $8n_{Q4}^3 + 13n_{Q4}^2 = \gamma(8n_{Q2}^3 + 13n_{Q2}^2)$. Combining these we can solve for the correct number of cores (PEs) to achieve the outcome, $k_{Q4}[cores] = \gamma k_{Q2}[cores]$, as well as solve

1 process , dgemm() C <--- a A B + b C
OPERATIONAL COMPLEXITY :A[m,n] , B[n,p] , C[m,p] :: [2mpn + 2mp] FLOP
E.g. m=n=p=1024 ---> 2149580800 FLOP , 1 measure 2156920832

N	1024	2048	4096	8192
T[s]	0.314	2.548	21.978	174.85
INS	1717503641	13700971039	1.09391E+11	8.7426E+11
FLOP	2156920832	17251172352	1.37993E+11	1.10387E+12

1 process , zgemm() C <--- a A B + b C
OPERATIONAL COMPLEXITY :A[m,n] , B[n,p] , C[m,p] :: [8mpn + 13mp] FLOP
E.g. m=n=p=1024 ---> 8603566080 FLOP , 1 measure 8639217664

N	1024	2048	4096	8192
T[s]	1.238	9.999	79.615	
INS	6871120592	54839796617	4.38202E+11	
FLOP	8639217664	69084381184	5.52558E+11	

6 processes [2x3] nb=80 , pzgemm()

N	1024	2048	4096	8192	16384 (8pe)
T[s]	0.340	2.224	15.904	124.33	686.7
INS	9497229846	66454914633	4.97548E+11	3.88845E+12	3.06305E+13
FLOP	8932819064	71433191544	5.71348E+11	4.57031E+12	3.65606E+13

Figure 3: The figure presents a check of the PAPI software tool on the target platform. The operation is the BLAS 3 kernel for the general product of matrices, $C \leftarrow \alpha AB + \beta C$. The theoretical complexity of the operation is stated as well as the total instructions and floating point instruction counts collected directly with PAPI for a handful of instances. The agreement of the measured counts with our theoretical estimate is outstanding.

the cubic polynomial equation $8n_{Q4}^3 + 13n_{Q4}^2 - constant = 0$ for n_{Q4} to obtain the approximate problem size where $constant := \gamma(8n_{Q2}^3 + 13n_{Q2}^2)$ and n_{Q2} is chosen by the user. Thus, we know how to initialize the parameter of our problem in both Q2 and Q4 to achieve the scaling result as projected onto the linear scaling model.

It is concluded that the tool provides a powerful instrument for understanding the interaction of our software with the system. It remains the case that the benefit of increased computing power must be interpreted as to its value from the perspective of the problem domain.

Computational Science Capability: DCA++

0.3 Introduction

0.4 Background and Motivation

Recent experiments have shown that nanoscale charge and spin inhomogeneities emerge in a number of cuprates. Based on these findings, it was proposed in the literature that inhomogeneities play a major role in high-temperature superconductivity. The results of DCA++ calculations will be usefull for studying the role of inhomogeneities in the pairing mechanism of the 2D Hubbard model and answer such questions as: Do inhomogeneities act to increase or decrease the critical temperature Tc? Do they enhance, suppress or even modify the pairing mechanism? Is there an optimal inhomogeneity that maximizes Tc? Technologically, especially the last question is of great importance since one could potentially use the knowledge gained from these runs to artificially structure cuprate based materials with higher transition temperatures. Nummerical studies of disordered systems will help to clarify one of the most important and timely questions in high-temperature superconductivity research, and thus will have high impact in this field.

0.5 Capability Overview

0.5.1 Physical Model

The essential features of the cuprates were recognized soon after their discovery in 1986 and summarized in a paper by Anderson in 1987 [Anderson1987]: (1) the structural units are the two-dimensional CuO_2 planes (Fig. 4a), and (2) the superconductivity is created by doping these planes that would otherwise be a Mott insulator with charge carriers (electrons or holes). Zhang and Rice pointed out [Zhang1988] that due to the strong on-site coulomb repulsion, these carriers form a localized singlet state consisting of the Cu $d_{x^2-y^2}$ orbital hybridized with the respective p_x and p_y orbitals of the neighboring O atoms (Fig. 4b). These considerations lead to a remarkably simple model description, the two-dimensional (2D) Hubbard Model, in which the carriers can hop between their nearest neighbor sites on a square lattice with transition ampli-

tude t and interact through an onsite Coulomb repulsion U (Fig. 4c). Due to the strength of the Coulomb repulsion ($U \sim 5-10\,eV \gtrsim 8t$), the sites will be mostly singly occupied, hence leading to the formation of atomic moments due to the spin 1/2 character of the carriers. Furthermore, because of the Pauli exclusion principle for Fermions, which precludes double occupation of a site with two carriers of the same spin orientation, the magnetic moments on neighboring sites will tend to order antiferromagnetically in order to minimize the kinetic energy.

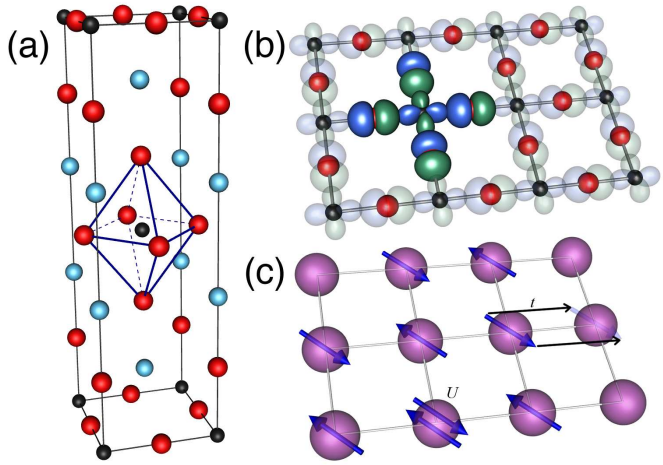

Figure 4: (a) The crystal structure of La_2CuO_4, a typical cuprate, where black, red, and blue spheres represent Cu, O, and La, respectively. (b) The CuO_2 plane with outlines of the Cu $d_{x^2-y^2}$ and O p_x and p_y orbitals. Also shown in full color is the Zhang-Rice singlet state that forms from hybridization of the Cu orbitals with the neighboring O orbitals (see text). (c) Pictorial representation of the single band 2D Hubbard model with on- site Coulomb repulsion U and inter-site hopping t.

Our simulations employ a quantum cluster method known as the Dynamical Cluster Approximation (DCA) [Hettler1998,Maier2005] in conjunction with the Hirsch-Fye QMC (HF-QMC) algorithm [Hirsch1986]. The approach in these simulations is to solve the quantum many-body problem at the atomic and nano-scale exactly with QMC, and to account for the macroscopic length scales within a mean-field approximation by self-consistently and coherently embedding the cluster solution into an effective medium. In the limit of infinitely large clusters, the DCA/QMC recovers the exact solution.

The DCA++ code implements the DCA/QMC and other quantum cluster methods in the framework of generic programming techniques and allows for effective extensions of the models. In particular, here we focus on an extension of the model that will allow us to study models with random disorder or other types of nano-scale variations of the effective coulomb interaction. These extensions are motivated by recent experiments that have found spatial variations of T_c and even identified local regions in which the signatures of su-

perconductivity (superconducting gap in the excitation spectrum) persisted to temperatures well above the macroscopically measured T_c [Gomes2007]. With new and much larger supercomputers, a systematic study of the disorder effects on large clusters will be possible.

0.5.2 Numerical Method

A comprehensive review of quantum cluster methods and the DCA can be found in a recent review article [Maier2005], and a detailed description of the DCA/QMC algorithm has been given in reference [Jarrell2001]. We limit our discussion of the DCA/QMC method here to a brief overview.

We are interested here in the 2D Hubbard model with atomic scale disorder in the Coulomb interaction parameter U. The Hamiltonian operator of a particular distribution ν of onsite interaction $\{U_i^{(\nu)}\}$ is conveniently written in second quantized notation

$$H^{(\nu)} = -\sum_{ij\sigma} t_{ij} c_{i\sigma}^{\dagger} c_{j\sigma} + \sum_i U_i^{(\nu)} n_{i\uparrow} n_{i\downarrow}, \tag{6}$$

where the operator $c_{i\sigma}^{\dagger}$ ($c_{j\sigma}$) creates (annihilates) a particle with spin σ on site i, t_{ij} is the hopping amplitude between sites, and $n_{i\sigma} = c_{i\sigma}^{\dagger} c_{i\sigma}$ is the density operator. The first part of this Hamiltonian is usually refered to as the non-interacting part, $H_0 = -\sum_{\mathrm{ij}\,\sigma} t_{\mathrm{ij}} c_{i\sigma}^{\dagger} c_{j\sigma}$.

The basic idea in DCA is that quantum correlations only have to be treated explicitly within a cluster of N_c atoms, and that all correlations outside this cluster can be treated in a mean-field way, that is, the system outside the cluster is represented by an effective medium. In this approximation, the self-energy is fully described on the set of *coarse grained* k-points K, which correspond to the lattice Fourier transform of the cluster. There are N_c such k-points in the Brillouin zone and we introduce $M(k)$, a function that maps every k-point of the Brillouin zone onto the nearest coarse grained point K. Within the DCA, and assuming that the self-energy $\Sigma(K, z)$ can be somehow computed, the Green function can be written as

$$G(k, z) = \left[z - \epsilon_0(k) - \Sigma(M(k), z)\right]^{-1}. \tag{7}$$

In practice, this function is represented on a discrete but dense uniform mesh of N k-points in the Brillouin zone. In the first iteration of the DCA/QMC algorithm, one usually starts out by setting Σ to zero or by using a self-energy that has been computed with a computationally inexpensive approximation (such as second order perturbation theory). One now proceeds by computing the Coarse Grained Green function

$$\overline{G}(K, z) = \frac{N_c}{N} \sum_{k=K+\tilde{k}} \left[z - \epsilon_0(K + \tilde{k}) - \Sigma(K, z)\right]^{-1}, \tag{8}$$

and with it computes the cluster excluded Green function of the effective medium

$$G_0'(K, z) = \left[\overline{G}^{-1}(K, z) + \Sigma(K, z)\right]^{-1}. \tag{9}$$

Note that this cluster excluded Green function is only equal to the non-interacting Green function $G_0(k,z)$ of the previous subsection in the limit of an infinite cluster size, i.e., when $N_c = N \rightarrow \infty$ and the self-energy is exact. However, if we lattice Fourier transform $G'_0(K,z)$ into real space, the resulting real space cluster excluded Green function $G'_0(X_i - X_j, z)$ plays the role of the non-interacting real space Green function of the embedded cluster $[z - \tilde{H}_0(X_i, X_j; z)]^{-1}$. (Note that when we solve the quantum many-body problem on the cluster the $\{X_i\}$ denote the sites of the cluster and are therefore a subset of $\{x_i\}$). Here, $\tilde{H}_0(X_i, X_j; z)$ describes the dynamics of the electrons on the non-interacting cluster and their time-dependent excursions into the effective medium. The next step in the DCA algorithm is to use an adequate quantum cluster solver to determine the cluster Green function $G_c^{(\nu)}(X_i, X_j, z)$ for a particular configuration $\{U_i^{(\nu)}\}$ of on-site Coulomb repulsions. The configurationally averaged cluster Green function is given by

$$G_c(X_i - X_j, z) \quad = \quad \frac{1}{N_d} \sum_{\nu=1}^{N_d} G_c^{(\nu)}(X_i, X_j, z). \tag{10}$$

Note that because the cluster excluded Green's function is the lattice Fourier transform and since we average over all configurations (for example if we have n possible values of U per site we would have $N_d = n^{N_c}$ configurations $\{U_i^{(\nu)}\}$), the cluster excluded Green function will have periodic boundary conditions and can be lattice Fourier transformed into $G_c(K,z)$. The new estimate to the self energy is given by

$$\Sigma(K,z) \quad = \quad G_0^{-1}(K,z) - G_c^{-1}(K,z), \tag{11}$$

and can be used to compute a new cluster excluded Green function with equation (9). The DCA method is iterated until the self energy converges. With a converged self-energy, one can compute the Green function $G(k,z)$ of the system as well as many other quantities needed to analyze the solutions of the Hubbard model.

Quantum cluster methods such as the DCA map the problem onto an effective cluster self-consistently embedded in a mean-field. The effective cluster problem is solved with a massively parallel Hirsch-Fye quantum Monte Carlo (QMC) algorithm. Along the QMC Markov chain, measurements of physical quantities such as the single-particle Greens function and two-particle correlation functions are performed. Between the measurements, the Greens function is updated using a Dyson equation. The majority of the CPU time of a typical DCA QMC simulation is spent in the Greens function updates and the measurements. These two inner loops are highly optimized to run efficiently on the NCCS machines. The Greens function updates are given by a vector outer product, This computation is optimized by delaying the update, thus effectively replacing the vector outer product by a slender rectangular matrix-matrix multiply. This allows us to perform the Greens function update very efficiently with the BLAS level 3 call DGEMM. The other CPU intensive task is the measurement of two-particle correlation functions. In the QMC technique, this reduces to evaluating products of Greens functions which are optimized by transforming from space-time to reciprocal space and frequency. These Fourier transforms are handled using the BLAS level 3 call ZGEMM.

0.5.3 Software Implementation

The two inner loops in the Green's function updates have previously been optimized to run efficiently on the NCCS machines: The Greens function updates are given by a vector outer product, $Gc = Gc + a * b$, where the Greens functions Gc and Gc are matrices of size $Nt \times Nt$. To optimize this computation, we delay the update, thus effectively replacing the vector outer product by a slender rectangular matrix-matrix multiply for matrices of size $32 \times Nt$. This allows us to perform the Greens function update very efficiently with the BLAS level 3 call DGEMM. Previously, on the Cray X1E we experienced up to a 5-fold speedup as a result of the delayed updates.

The other CPU intensive task is the measurement of two-particle correlation functions. In the QMC technique, this reduces to evaluating products of Greens functions which are optimized by transforming from space-time to reciprocal space and frequency. These Fourier transforms are handled using the BLAS level 3 call ZGEMM and therefore run at speeds near peak performance.

The QMC algorithm is parallelized in the standard way for Monte Carlo methods by distributing the Markov chain onto many processors. Several independent, shorter Markov-chain walks on different processors are performed and the final result is obtained by averaging the results of each walk using MPI. Apart from the fraction of the walk required to achieve equilibrium, the result is an almost perfectly parallel speedup with an increasing number of processors. This arises because no communication is required in the inner loops of the code. As a result, the code scales to several hundreds of processing units with almost ideal speedup.

The DCA++ code is written in the C++ programming language using generic programming models like the C++ Standard Template Library and the Psi-Mag toolkit. BLAS libraries are called for dense linear algebra computations in the inner loops of the code. The MPI library is used for parallelization and communication.

0.5.4 Execution Performance

0.5.5 References

1. P. W. Anderson. The resonating valence bond state in la_2cuo_4 and superconductivity. Science 235:1196, 1987.

2. F. Zhang, T. and Rice. Effective hamiltonian for the superconducting Cu oxides. Phys. Rev. B 37:R3759, 1988.

3. J. Hirsch, and R. Fye. Monte carlo method for magnetic impurities in metals. Phys. Rev. Lett. 56:2521, 1986.

4. M. H. Hettler, A. N. Tahvildar-Zadeh, M. Jarrell, T. Pruschke, and H. R. Krishnamurthy. Non- local dynamical correlations of strongly interacting electron systems. Phys. Rev. B, 58:R7475 R7479, 1998.

5. M. H. Hettler, M. Mukherjee, M. Jarrell, and H. R. Krishnamurthy. Dynamical cluster ap- proximation: Nonlocal dynamics of correlated electron systems. Phys. Rev. B, 61:1273912756, 2000.

6. M. Jarrell, Th. Maier, M. H. Hettler, and A. N. Tahvildarzadeh. Phase diagram of the hubbard model: Beyond the dynamical mean field. Europhys. Lett., 56:563, 2001.

7. Th. Maier, M. Jarrell, Th. Pruschke, and J. Keller. A non-crossing approximation for the study of intersite correlations. Eur. Phys. J B, 13:613, 2000.

8. Th. Maier, M. Jarrell, Th. Pruschke, and J. Keller. d-wave superconductivity in the hubbard model. Phys. Rev. Lett., 85:1524, 2000.

9. Th. Maier, M. Jarrell, Th. Pruschke, and M.H. Hettler. Quantum cluster theories. Rev. Mod. Phys., 77:1027, 2005.

10. M. Jarrell, Th. Maier, C. Huscroft, and S. Moukouri. A quantum monte carlo algorithm for non-local corrections to the dynamical mean-field approximation. Phys. Rev. B, 64:195130, 2001.

11. K. K. Gomes, A. N. Pasupathy, A. Pushp, S. Ono, Y. Ando, and A. Yazdani. Nature 447, 2007.

0.6 Metric Problem

0.6.1 Intent

Present understanding of physical effects contributing to the behavior of high-Tc superconductors require the investigation of models including disorder. The total number of disorder configurations N_{dis} is proportional to 2^{N_c}, where N_c is the number of cluster sites. The metric is intended to show the scaling of the disorder configurations that are accessible during a fixed wall clock time with the number of available compute nodes. The problem chosen here is representative of production runs, but the number of Green's function measurements has been reduced to a number that allows the timing runs to finish in approximatly one hour of wall clock time. The ability to weakly scale to larger numbers of disorder representations will enable us to obtain the disorder dependence of the superconducting transition temperature with increasingly better statistics as the available machines grow in size.

0.6.2 Model parameters

We choose the following parameters for the Hubbard model to be investigated in these scaling investigations:

- Number of sites: $N_c = 16$
- Number of time slices: $N_t = 150$
- Hubbard U: $U \in \{0.5, 1.5\}$ randomly chosen for a disorder configuration.
- Number of disorder configurations: $N_{dis} = 16$
- Number of selfconsistent measurements: 20

- number of warm-up steps: 40

The metric to establish the scaling to larger disorder counts will keep all the input parameters to the probem fixed and only vary N_{dis} and determine the CPU-time per disorder configuration t_{dis}.

$$t_{dis} = \frac{N_{CPU} \times t_{wall}}{N_{dis}}$$

Where N_{CPU} is the total number of MPI processes/cores for the run, t_{wall} is the wall clock time for the run as return by calls to `PAPI_get_real_usec()` at the begining and end of the DCA++ main routine. During the time measurements the calculation of self energies was excluded since this is a fixed amount of computation independent of the number of selfconsistent measurements and a production run will require ≈ 10 times the measuremts performed during the benchmark runs.

0.7 Metric Baseline

0.7.1 Results and Interpretation

The code was run with the model parameters described above on April 3, 2008 on 1956 Opteron quad core nodes on the NCCS jaguar system with one MPI process per core. This results in a total of $N_{CPU} = 7824$ MPI processes and 489 processes pre disorder configuration. The runtime for the code returned by `PAPI_get_real_usec` was $t_{wall} = 3336.22\text{s} = 55\text{min } 36.22\text{s}$. These number result in

$$t_{dis}^{\text{Baseline}} = 1631411.58\text{s} = 453\text{h } 10\text{min } 11.58\text{s}.$$

In addition PAPI was used to collect the number of real cycles N_{cyc}, and the folloing events:

- `PAPI_TOT_INS` The total number of instructions N_{inst}
- `PAPI_FP_OPS` The total number of floating point operations N_{FP}
- `RETIRED_SSE_OPERATIONS:DOUBLE_ADD_SUB_OPS:DOUBLE_MUL_OPS:DOUBLE_DIV_OPS:OP_TYPE` The number of double precission floating point operations N_{dFP}
- `RETIRED_SSE_OPERATIONS:SINGLE_ADD_SUB_OPS:SINGLE_MUL_OPS:SINGLE_DIV_OPS:OP_TYPE` The number of single precission floating point operations N_{sFP}

We accumulate these counters over the runtime of the code and over all MPI processes and find

The number of single precission operations in the code is identically zero, as expected, since all calculations are performed in double precision. Currently we are in the process of identifying which parts of the code could benefit from using single precission arithmetic to reduce the computation time required. With these counters we can establish the performance of the code by dividing by the run time. This results in an execution speed of $N_{inst}/t_{wall} = 21.413 \times 10^{12}\text{inst/s}$ or $2.737 \times 10^{9}\text{inst/s}$ per core and a floating point performance of $N_{FP}/t_{wall} = 19.785 \times 10^{12}\text{FLOP/s}$ or 2.529GFLOP/s per core which

Table 10: results ... results ... results

	N_D	N_{cores}	N_{cyc}	N_{inst}	N_{FP}	t_{user}
$\beta = 90$ – Q2	64	7,808	1.0725×10^{17}	1.2290×10^{17}	1.0372×10^{17}	6,540
$\beta = 120$ – Q2	64	7,808	2.3658×10^{17}	2.7490×10^{17}	2.4029×10^{17}	14,427
$\beta = 150$ – Q2	64	7,808	$XX \times 10^{17}$	$XX \times 10^{17}$	$XX \times 10^{17}$	XX
$\beta = 90$ – Q4	256	31,232	4.2347×10^{17}	4.7357×10^{17}	4.0821×10^{17}	6,484
$\beta = 120$ – Q4	256	31,232	9.1204×10^{17}	1.0397×10^{17}	9.4308×10^{17}	13,946
$\beta = 150$ – Q4	256	31,232	1.5590×10^{18}	1.9300×10^{18}	1.8126×10^{18}	23,791

corresponds to 30.1% of the theoretical peak of 8.4GFLOP/s (4 floating point operations/cycle at 2.1GHz).

This scaling enabled us, in further calculations not benchmarked here, to investigate the transition to a superconducting state with d-wave symmetry. The order parameter that signals a transition to this state is given by

$$\Delta_d^\dagger = \sum_{\mathbf{k}} g(\mathbf{k}) c^\dagger_{\mathbf{k},\uparrow} c^\dagger_{-\mathbf{k},\downarrow} \,, \tag{12}$$

where $g(\mathbf{k}) = \cos \mathbf{k}_x - \cos \mathbf{k}_y$ is a d-wave form-factor. In linear response theory, the pair-field susceptibility formed from this order parameter

$$P_d = \int_0^\beta d\tau \langle T_\tau \Delta_d(\tau) \Delta_d^\dagger(0) \rangle \tag{13}$$

diverges at the transition temperature T_c to the d-wave superconducting state. Using the Bethe-Salpeter equation, the pair-field susceptibility may also be written in terms of a particle-particle vertex function Γ^{pp}

$$P_d = P_d^0 + P_d^0 \Gamma^{pp} P_d \,. \tag{14}$$

Here, P_d^0 and Γ^{pp} are matrices of size $NL \times NL$, where N is the number of lattice sites and L the number of Matsubara-frequencies, P_d^0 is the pair-field susceptibility of the non-interacting part of the model and the right hand side contains an implicit sum over the matrix elements. In the DCA, the vertex function Γ^{pp} is approximated by its corresponding cluster quantity which is a smaller matrix of size $N_c L \times N_c L$. This matrix is calculated from the cluster pair-field susceptibility which is measured in the HF-QMC process in the last DCA iteration. Since Eq. (14) can be written as

$$P_d = \frac{P_d^0}{1 - \Gamma^{pp} P_d^0} \,, \tag{15}$$

one can conveniently determine instabilities by calculating the eigenvalues and eigenvectors of the pairing matrix $\Gamma^{pp} P_d^0$, i.e., solving

$$-\frac{T}{N_c} \sum_{K'} \Gamma^{pp}(K, K') P_d^0(K') \phi_\alpha(K') = \lambda_\alpha \phi_\alpha(K) \,. \tag{16}$$

The susceptibility diverges when the leading eigenvalue λ_α crosses one which determines T_c. The symmetry of the ordered state is then given by the K dependence of the corresponding eigenvector $\phi_\alpha(K)$.

At low temperatures the leading eigenvector has $d_{x^2-y^2}$-symmetry. The leading eigenvalue is shown as a function of temperature in Fig. 5

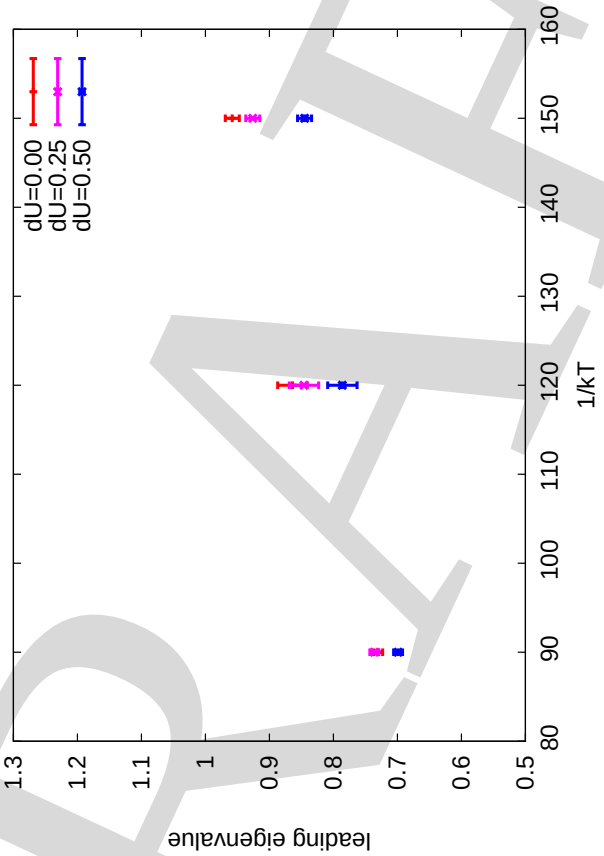

Figure 5: The leading eigenvalue λ_d of the Bethe-Salpeter equation, Eq. (16), calculated on a 16-site cluster for $U_i = 4t(1 + \xi_i dU)$, where ξ_i is a random number with value ± 1 for three different values of disorder.

In the system without disorder ($dU = 0$), the leading eigenvalue crosses one at the transition temperature $T_c^{\text{clean}} = 0.08t$. In the disordered system, T_c is reduced, although the reduction is only significant for $dU = 0.5$. One sees that as the disorder strength is increased, T_c decreases. It therefore must be concluded that T_c is suppressed by disorder in the interaction strength.

0.7.2 Baseline input

```
#geometry 2d
```

```
#SizeOfTheCluster  4,2,0,4,640000
#NumberOfTimeSlices  90
#SizeOfUpdateBlock  32
#InverseTemperature   90.
#outputfilename output.A
#Ntr 16
#HubbardUType random
#HubbardUConfigs  64
#HubbardUCentral 1.0
#HubbardUDelta   0.5
#BareOrbitalEnergy  0.16910238
#Tprime 0.0
#NumberOfMatsubaras 800
#FreqType complex
#IsDisorderPresent 0
#SelfConsistentIterations 1
#SelfConsistentWarmUps 40 40
#SelfConsistentSkip 4
#MaxSkip 40
#SelfConsistentMeas   300
#Bins   122
#iauto 2
#SelfConsistentConvergence 0.75
#DcaMinError -1
#ifilter 0
#icond 1
#iosft 1
#density 0.90
#AdjustDensity 1
#npt 1
#initsigma filesigma_beta90.dat
#initdelta 0.001
#usedgemv 1
#usedgemm 1


#geometry 2d
#SizeOfTheCluster  4,2,0,4,640000
#NumberOfTimeSlices  120
#SizeOfUpdateBlock  32
#InverseTemperature   120.
#outputfilename output.A
#Ntr 16
#HubbardUType random
#HubbardUConfigs  64
#HubbardUCentral 1.0
#HubbardUDelta   0.5
#BareOrbitalEnergy  0.16809740858455499
#Tprime 0.0
#NumberOfMatsubaras 800
#FreqType complex
```

```
#IsDisorderPresent 0
#SelfConsistentIterations 1
#SelfConsistentWarmUps 40 40
#SelfConsistentSkip 4
#MaxSkip 40
#SelfConsistentMeas   300
#Bins   122
#iauto 2
#SelfConsistentConvergence 0.75
#DcaMinError -1
#ifilter 0
#icond 1
#iosft 1
#density 0.90
#AdjustDensity 1
#npt 3
#initsigma filesigma_beta120.dat
#initdelta 0.001
#usedgemv 1
#usedgemm 1

#geometry 2d
#SizeOfTheCluster  4,2,0,4,640000
#NumberOfTimeSlices  150
#SizeOfUpdateBlock  32
#InverseTemperature   150.
#outputfilename output.A
#Ntr 16
#HubbardUType random
#HubbardUConfigs  64
#HubbardUCentral 1.0
#HubbardUDelta   0.50
#BareOrbitalEnergy  0.16791218624384005
#Tprime 0.0
#NumberOfMatsubaras 800
#FreqType complex
#IsDisorderPresent 0
#SelfConsistentIterations   1
#SelfConsistentWarmUps 40 40
#SelfConsistentSkip 4
#MaxSkip 40
#SelfConsistentMeas   300
#Bins   122
#iauto 2
#SelfConsistentConvergence 0.75
#DcaMinError -1
#ifilter 0
#icond 1
#iosft 1
#density 0.90
#AdjustDensity 1
```

```
#npt 3
#initsigma filesigma_beta150.dat
#initdelta 0.001
#usedgemv 1
#usedgemm 1
```

0.7.3 Q4 runs input

Note that the twoparticle option specified for the Q4 runs was the default in the DCA++ code during the Q2 runs.

```
#options twoparticle
#geometry 2d
#SizeOfTheCluster  4,2,0,4,640000
#NumberOfTimeSlices  90
#SizeOfUpdateBlock  32
#InverseTemperature   90.
#outputfilename output.B
#Ntr 16
#HubbardUType random
#HubbardUConfigs  256
#HubbardUCentral 1.0
#HubbardUDelta   0.5
#BareOrbitalEnergy  0.16910238
#Tprime 0.0
#NumberOfMatsubaras 800
#FreqType complex
#IsDisorderPresent 0
#SelfConsistentIterations 1
#SelfConsistentWarmUps 40 40
#SelfConsistentSkip 4
#MaxSkip 40
#SelfConsistentMeas   300
#Bins   122
#iauto 2
#SelfConsistentConvergence 0.75
#DcaMinError -1
#ifilter 0
#icond 1
#iosft 1
#density 0.90
#AdjustDensity 1
#npt 1
#initsigma filesigma.1.A
#initdelta 0.001
#usedgemv 1
#usedgemm 1

#options twoparticle
#geometry 2d
#SizeOfTheCluster  4,2,0,4,640000
```

```
#NumberOfTimeSlices  120
#SizeOfUpdateBlock  32
#InverseTemperature   120.
#outputfilename output.B
#Ntr 16
#HubbardUType random
#HubbardUConfigs  256
#HubbardUCentral 1.0
#HubbardUDelta   0.5
#BareOrbitalEnergy  0.16809740858455499
#Tprime 0.0
#NumberOfMatsubaras 800
#FreqType complex
#IsDisorderPresent 0
#SelfConsistentIterations 1
#SelfConsistentWarmUps 40 40
#SelfConsistentSkip 4
#MaxSkip 40
#SelfConsistentMeas   300
#Bins   122
#iauto 2
#SelfConsistentConvergence 0.75
#DcaMinError -1
#ifilter 0
#icond 1
#iosft 1
#density 0.90
#AdjustDensity 1
#npt 3
#initsigma filesigma.1.A
#initdelta 0.001
#usedgemv 1
#usedgemm 1


#options twoparticle
#geometry 2d
#SizeOfTheCluster  4,2,0,4,640000
#NumberOfTimeSlices  150
#SizeOfUpdateBlock  32
#InverseTemperature   150.
#outputfilename output.B
#Ntr 16
#HubbardUType random
#HubbardUConfigs  256
#HubbardUCentral 1.0
#HubbardUDelta   0.50
#BareOrbitalEnergy  0.16791218624384005
#Tprime 0.0
#NumberOfMatsubaras 800
#FreqType complex
```

```
#IsDisorderPresent 0
#SelfConsistentIterations   1
#SelfConsistentWarmUps 40 40
#SelfConsistentSkip 4
#MaxSkip 40
#SelfConsistentMeas   300
#Bins   122
#iauto 2
#SelfConsistentConvergence 0.75
#DcaMinError -1
#ifilter 0
#icond 1
#iosft 1
#density 0.90
#AdjustDensity 1
#npt 3
#initsigma filesigma.1.A
#initdelta 0.001
#usedgemv 1
#usedgemm 1
```

Computational Science Capability: GYRO

0.8 Introduction

0.9 Background and Motivation

The most promising and aggressively studied concept for power production by fusion reactions is the tokamak. At the present time, magnetic fusion energy research has reached the point where construction of a tokamak burning plasma facility is prudent. To this end, on 21 November 2006, seven participants (the European Union, India, Japan, People's Republic of China, Russia, South Korea, and the USA) formally agreed to fund ITER [**?**]. The ITER project, based in Cadarache, France, is anticipated to last for 30 years (10 for construction and 20 for operation) with the first plasma expected in 2016. ITER will be designed to produce approximately 500MW of fusion power sustained for up to 400 seconds. To achieve this design target, thermonuclear heating must balance transport and radiation losses. However, despite the advances made in the understanding and control of tokamak plasmas, various theoretical and technical uncertainties remain in reliably predicting confinement properties and performance of a reactor-scale devices. Within the worldwide fusion community, it is widely agreed that the gyrokinetic-Maxwell (GKM) equations [**?**, **?**] provide a solid foundation for the first-principles calculation of turbulent tokamak heat and particle transport. For years, the numerical solution of the nonlinear GKM equations has been a computational physics "Grand Challenge". Development of GYRO was partially funded by the *Plasma Microturbulence Project*, a fusion SciDAC project.

Traditionally, long-wavelength, low-frequency turbulence driven by ion-scale instabilities (ion-temperature-gradient and trapped-electron modes) is studied separately from the short-wavelength, high-frequency turbulence driven by electron-scale instabilities (electron-temperature-gradient modes). Recently, high-resolution, massively-parallel, multi-scale simulations with GYRO have shown that critical aspects of the physics can be missed or determined incorrectly if the electron and ion physics is not properly coupled. For example, it was found that a popular simplified model of ion physics previously used in studies of electron-scale turbulence can lead to nonphysical runaway of electron heat transport. This nonphysical runaway is eliminated when correct long-wavelength ion physics is self-consistently included. In addition, GYRO

multi-scale simulations showed that under normal conditions, most of the electron heat transport arises from large-scale instabilities. However, when these large-scale instabilities are suppressed by plasma rotation or other processes, the electron instabilities survive and may dominate the loss of electron heat from the plasma. A further remarkable finding is that strong turbulence at long scales can act to reduce the intensity of turbulence at short scales. The simulations which led to these discoveries were carried out on the Cray X1E computer at ORNL, with the largest runs taking about a week on 720 multi-streaming processors.

0.10 Capability Overview

The GYRO code, written primarily in Fortran 90, solves the nonlinear GKM equations for core ions, electrons and any number of plasma impurity species. Electromagnetic or reduced electrostatic simulations are possible. The computational domain can be radially global, with input data taken from an experimental database, or radially local, in order to accurately study isolated parametric effects. Both partial and full torus simulations are possible, although the latter is rarely (if ever) necessary to obtain a converged result. GYRO uses a five-dimensional Eulerian grid and advances the system in time using a second-order, implicit-explicit (IMEX) Runge-Kutta (RK) integrator [**?**]. There is also an option to use a fourth-order explicit Runge-Kutta time advance.

0.10.1 Physical Model

The Gyrokinetic Model

The plasma configuration is represented by the single-particle kinetic distribution f_σ, where σ is a species label, and is written as a sum of an equilibrium part, $F_{0\sigma}$, and fluctuating terms:

$$f_\sigma(\mathbf{x}, E, \mu, t) = F_{0\sigma}(\mathbf{x}, E) + \delta f_\sigma(\mathbf{x}, E, \mu, t) \ , \tag{17}$$

where the perturbed distribution, δf_σ is given by

$$\delta f_\sigma(\mathbf{x}, E, \mu, t) = -\frac{z_\sigma e}{T_\sigma} F_{0\sigma} \left[\delta\phi(\mathbf{x}, t) - \mathcal{G}_\sigma \delta\phi(\mathcal{R}, t) + \frac{v_\parallel}{c} \mathcal{G}_\sigma \delta A_\parallel(\mathcal{R}, t) \right] + h_\sigma(\mathcal{R}, E, \mu, t) \ . \tag{18}$$

Above, $\mathbf{x} = \mathcal{R} + \boldsymbol{\rho}$ is the particle position, $\boldsymbol{\rho} = \hat{\mathbf{b}} \times \boldsymbol{v}/\Omega_{c\sigma}$ is the gyroradius vector, $\Omega_{c\sigma} = z_\sigma e B/(m_\sigma c)$, $\mathcal{R}$ is the guiding-center position, $E_\sigma = m_\sigma v^2/2$ is the energy, and $\mu \doteq v_\perp^2/(2B)$ is the magnetic moment per unit mass. The symbol $\mathcal{G}_\sigma$ denotes a gyroaverage, which can be defined formally as

$$\mathcal{G}_\sigma z(\mathcal{R}, t) \doteq \oint \frac{da}{2\pi} \, z(\mathcal{R} + \boldsymbol{\rho}(a), t) \ , \tag{19}$$

for any function, z. Fundamental considerations required that the equilibrium is a local Maxwellian

$$F_{0\sigma}(\mathcal{R}, E) \doteq \frac{n_\sigma}{(2\pi T_\sigma/m_\sigma)^{3/2}} e^{-E/T_\sigma} = n_\sigma \, F_{M\sigma} \ . \tag{20}$$

In the expression for $F_{0\sigma}$, we have ignored a factor of $\exp(z_\sigma e\Phi_0/T_\sigma)$. In connection with the equilibrium potential, we assume $\psi\, d\Phi_0/d\psi \gg \Phi_0$ in order to perturbatively account for $\mathbf{E}\times\mathbf{B}$ shearing effects. For normalization purposes it is useful to note that

$$\int d^3v\, F_{M\sigma} = 1\ . \tag{21}$$

Magnetic topology

In what follows we adopt the right-handed, field-aligned coordinate system (ψ,θ,α) together with the Clebsch representation [**?**] for the magnetic field

$$\mathcal{B} = \nabla\alpha\times\nabla\psi \quad \text{such that} \quad \mathcal{B}\cdot\nabla\alpha = \mathcal{B}\cdot\nabla\psi = 0\ . \tag{22}$$

The angle α is written in terms of the toroidal angle φ as

$$\alpha \doteq \varphi + \nu(\psi,\theta)\ . \tag{23}$$

In Eqs. (22) and (23), ψ (as we will show) is poloidal flux divided by 2π, and θ refers simultaneously to (a) an angle in the poloidal plane (at fixed φ), or (b) a parameterization of distance along a field line (at fixed α). In these coordinates, the Jacobian is

$$\mathcal{J}_\psi \doteq \frac{1}{\nabla\psi\times\nabla\theta\cdot\nabla\alpha} = \frac{1}{\nabla\psi\times\nabla\theta\cdot\nabla\varphi}\ . \tag{24}$$

Since the coordinates (ψ,θ,α) and (ψ,θ,φ) form right-handed systems, the Jacobian $\mathcal{J}_\psi$ is positive-definite. In the latter coordinates, the magnetic field becomes

$$\mathcal{B} = \nabla\varphi\times\nabla\psi + \frac{\partial\nu}{\partial\theta}\nabla\theta\times\nabla\psi \tag{25}$$

Using the definition of the safety factor, $q(\psi)$, we may deduce

$$q(\psi) \doteq \frac{1}{2\pi}\int_0^{2\pi}\frac{\mathcal{B}\cdot\nabla\varphi}{\mathcal{B}\cdot\nabla\theta}\,d\theta = \frac{1}{2\pi}\int_0^{2\pi}\left(-\frac{\partial\nu}{\partial\theta}\right)d\theta = \frac{\nu(\psi,0)-\nu(\psi,2\pi)}{2\pi}\ . \tag{26}$$

For concreteness, we choose the following boundary conditions for ν:

$$\nu(\psi,2\pi) = -2\pi\, q(\psi)\ , \tag{27}$$

$$\nu(\psi,0) = 0\ . \tag{28}$$

By writing $\mathcal{B}$ in the standard form for up-down symmetric equilibria,

$$\mathcal{B} = \nabla\varphi\times\nabla\psi + f(\psi)\nabla\varphi\ , \tag{29}$$

we can derive the following integral for ν:

$$\nu(\psi,\theta) = -f(\psi)\int_0^\theta \mathcal{J}_\psi\,|\nabla\varphi|^2\,d\theta\ . \tag{30}$$

We remark that in the case of concentric (unshifted) circular flux surfaces, one will obtain the approximate result $\nu(\psi,\theta)\sim -q(\psi)\theta$.

Gyrokinetic equation

The gyrokinetic equation for h_σ can be written compactly as

$$\frac{\partial h_\sigma}{\partial t} + (v_\parallel \hat{\mathbf{b}} + \mathbf{v}_\mathrm{d}) \cdot \nabla H_\sigma + \delta\mathbf{v} \cdot \nabla \left(F_{0\sigma} + h_\sigma\right) + \mathbf{v}_{\mathrm{E}0} \cdot \nabla h_\sigma = C(h_\sigma) \,, \tag{31}$$

where the new symbols are

$$U = \delta\phi - \frac{v_\parallel}{c}\delta A_\parallel \,, \tag{32}$$

$$H_\sigma = h_\sigma + \frac{z_\sigma e}{T_\sigma} F_{0\sigma}\, \mathcal{G}_\sigma U \,, \tag{33}$$

and the the velocities are

$$\mathbf{v}_\mathrm{d} \doteq \frac{v_\parallel^2 + \mu B}{\Omega_{c\sigma} B^2}\, \mathcal{B} \times \nabla B + \frac{4\pi v_\parallel^2}{\Omega_{c\sigma} B^2} \hat{\mathbf{b}} \times \nabla p \tag{34}$$

$$\mathbf{v}_{\mathrm{E}0} \doteq \frac{c}{B}\, \hat{\mathbf{b}} \times \nabla \Phi_0 \,, \tag{35}$$

$$\mathbf{v}_\mathrm{E} \doteq \frac{c}{B}\, \hat{\mathbf{b}} \times \nabla \mathcal{G}_\sigma \delta\phi \,, \tag{36}$$

$$\delta\mathbf{v} \doteq \frac{c}{B}\, \hat{\mathbf{b}} \times \nabla \mathcal{G}_\sigma U = \mathbf{v}_\mathrm{E} + v_\parallel \frac{\delta \mathcal{B}_\perp}{B} \,. \tag{37}$$

Various terms can be simplified without refering to the geometry model. Within the accuracy of the gyrokinetic ordering, we have

$$\mathbf{v}_{\mathrm{E}0} \cdot \nabla h_\sigma = \frac{c}{B}\, \hat{\mathbf{b}} \times \nabla \Phi_0 \cdot \nabla h_\sigma \sim -c \frac{\partial h_\sigma}{\partial \alpha} \frac{\partial \Phi_0}{\partial \psi} \,, \tag{38}$$

$$\delta\mathbf{v} \cdot \nabla F_{0\sigma} = \frac{c}{B}\, \hat{\mathbf{b}} \times \nabla (\mathcal{G}_\sigma U) \cdot \nabla F_{0\sigma} \sim c \frac{\partial F_{0\sigma}}{\partial \psi} \frac{\partial (\mathcal{G}_\sigma U)}{\partial \alpha} \,, \tag{39}$$

$$\delta\mathbf{v} \cdot \nabla h_\sigma = \frac{c}{B}\, \hat{\mathbf{b}} \times \nabla (\mathcal{G}_\sigma U) \cdot \nabla h_\sigma \sim c \frac{\partial h_\sigma}{\partial \psi} \frac{\partial (\mathcal{G}_\sigma U)}{\partial \alpha} - c \frac{\partial h_\sigma}{\partial \alpha} \frac{\partial (\mathcal{G}_\sigma U)}{\partial \psi} \,. \tag{40}$$

Poisson equation

For a multi-species plasma the Poisson equation (including the Debye shielding term) can be written as

$$-\nabla_\perp^2 \delta\phi = 4\pi \sum_\sigma e z_\sigma\, \delta n_\sigma = 4\pi \sum_\sigma e z_\sigma \int d^3v\, \delta f_\sigma \,. \tag{41}$$

In terms of the function h_σ, we can write the expression above as

$$-\frac{1}{4\pi}\nabla_\perp^2 \delta\phi + \sum_\sigma n_\sigma \frac{e^2 z_\sigma^2}{T_\sigma} \int d^3v\, F_{M\sigma} \left(1 - \mathcal{G}_\sigma^2\right) \delta\phi = \sum_\sigma e z_\sigma \int d^3v\, \mathcal{G} h_\sigma \tag{42}$$

where σ runs over all species (ions and electrons).

Ampère equation

The primitive form of the Ampère equation is

$$\nabla_\perp^2 \delta A_\parallel = -\frac{4\pi}{c} \sum_\sigma \delta j_{\parallel,\sigma} = -\frac{4\pi}{c} \sum_\sigma e z_\sigma \int d^3v \, v_\parallel \, \delta f_\sigma \ . \tag{43}$$

Writing this in terms of h_σ gives

$$-\frac{1}{4\pi}\nabla_\perp^2 \delta A_\parallel + \sum_\sigma n_\sigma \frac{e^2 z_\sigma^2}{T_\sigma} \int d^3v \, \frac{v_\parallel^2}{c^2} \, F_{M\sigma} \, \mathcal{G}_\sigma^2 \delta A_\parallel = \sum_\sigma e z_\sigma \int d^3v \, \frac{v_\parallel}{c} \, \mathcal{G}_\sigma h_\sigma \ . \tag{44}$$

Illustrative form of the Model

The GKM equations have the following illustrative form, in which the fluctuating *gyrocenter* distribution f is coupled to the the electromagnetic fields, Φ:

$$\frac{\partial f}{\partial t} = \mathcal{L}_a f + \mathcal{L}_b \langle \Phi \rangle + \{ f, \langle \Phi \rangle \} \tag{45}$$

$$\mathcal{F}\Phi = \int\int dv_1 \, dv_2 \, \langle f \rangle \tag{46}$$

$\mathcal{L}_b$, $\mathcal{L}_b$ and $\mathcal{F}$ are linear operators, and $\langle \cdot \rangle$ is an operator which takes the average along a particle gyro-orbit. The nonlinearity, which has a Poisson bracket structure, appears in the gyrokinetic equation. The function $f(\mathbf{r}, v_1, v_2)$ is discretized over a 5-dimensional grid (three spatial and two velocity coordinates), while the 3-dimensional electromagnetic fields $\Phi(\mathbf{r}) = [\phi, A_\parallel]$ are independent of velocity. Here ϕ and $A_\parallel$ are the *electrostatic* and *electromagnetic* potentials, respectively. In order to obtain Eq. (45), one has averaged over the fast orbital motion (gyro-orbit) to eliminate the third velocity-space dimension (gyro-angle). However, this so-called *gyro-averaging* operation introduces *nonlocal* spatial operators, $\mathcal{F}$ and $\langle \cdot \rangle$, perpendicular to the magnetic field.

We remark that simulations normally reach a statistical steady state on a timescale much shorter than an energy confinement time. In practice, simulations are well into the steady-state regime after about 50,000 timesteps.

Geometry

The effects of cross-sectional shaping in GYRO are treated using the Miller local equilibrium model [**?**]. In the Miller model, nine dimensionless parameters are employed to describe the local equilibrium: κ (elongation), δ (triangularity), s (magnetic shear), R_0/a (aspect ratio), q (safety factor), $\partial R_0/\partial r$ (Shafranov shift), s_κ (elongation shear), s_δ (triangularity shear) and $\beta'_{\rm unit}$ (normalized pressure gradient). Note that

$$s_\kappa \doteq \frac{r}{\kappa}\frac{\partial \kappa}{\partial r} \ , \quad s_\delta \doteq r\frac{\partial \delta}{\partial r} \ , \quad \beta'_{\rm unit} \doteq -\frac{8\pi}{B_{\rm unit}^2}\frac{\partial p}{\partial r} \ . \tag{47}$$

The shape of a flux surface is specified using the following parameterization

$$R(r,\theta) = R_0(r) + r\cos\left(\theta + x \sin\theta\right) \ , \tag{48}$$

$$Z(r,\theta) = \kappa(r) r \sin\theta \tag{49}$$

where $x = \arcsin\delta$. The Miller algorithm constructs, via a simple computer program, a local equilibrium with the (R, Z) flux-surface shape given above. It is to be emphasized that said equilibrium is an exact solution of the Grad-Shafranov equation. A free parameter in this local equilibrium is the so-called *effective magnetic field strength*, $B_{\rm unit}(r)$ [**?**]. In practice, $B_{\rm unit}$ is determined only with reference to a global equilibrium through the relation

$$\frac{\partial\psi}{\partial r} = \frac{r}{q} B_{\rm unit} \ , \tag{50}$$

where r is the midplane minor radius and ψ is the toroidal flux divided by 2π. In a large-aspect-ratio, unshifted, circular plasma, $B_{\rm unit}(r) \to B_0$. The principal advantage of the Miller equilibrium model, in comparison to a full numerical equilibrium, is that the shape parameters can be individually varied, thus allowing for systematic studies of the isolated effects of each on the neoclassical transport. Furthermore, a given local equilibrium can be computed to high precision at low cost, free of spurious numerical artifacts often found in 2-D numerical equilibria. In the case of general geometry calculations, it is useful to introduce the *unit gyroradius*:

$$\rho_{a,{\rm unit}}(r) \doteq \frac{v_{ta}}{\Omega_{ca,{\rm unit}}} \quad \text{where} \quad \Omega_{ca,{\rm unit}}(r) = \frac{eB_{\rm unit}(r)}{m_a c} \ . \tag{51}$$

0.10.2 Numerical Method

Parallel Model

We briefly sketch the type of discretization scheme used in each dimension.

- $r \to i$ (radius): linear advective derivatives on f are treated with an upwind differences, whereas derivatives on fields are treated with centered differences. The nonlocal operators $\mathcal{F}$ and $\langle\cdot\rangle$ are approximated using a (banded) pseudospectral technique. The order of all discretizations is adjustable at run-time.

- $\tau \to j$ (poloidal angle): for f, there is no fixed grid in poloidal angle, θ. Instead, the transformation

$$\frac{v_\parallel(r,\lambda,\theta)}{R_0 q(r)} \frac{\partial}{\partial\theta} \to \Omega(r,\lambda) \frac{\partial}{\partial\tau} \tag{52}$$

  is used to eliminate the singularity at bounce points, $v_\parallel(\theta) = 0$. Here, $v_\parallel$ is the velocity parallel along the magnetic field, R_0 is the major radius of the torus, and q is the so-called *safety factor*. Then, an upwind scheme in τ is used to discretize $\partial f/\partial\tau$. The use of a τ-grid (leading to a different set of points in θ for every value of λ) for the GK equation dictates that the Maxwell equations are solved by expansion of fields in complex finite-elements: $\phi(r_i,\theta) = \sum_m F_m^i(\theta)$. The F_m^i satisfy a complex phase condition.

- $n_{\rm tor} \to n$ (toroidal angle): the toroidal direction (really, the direction perpendicular to both the radius, r, and to the magnetic field) is treated in a fully spectral manner. Note that simulations need not cover an entire toroidal circuit $(0, 2\pi]$. In fact, it is normally most efficient to cover a partial torus; for example: $n_{\rm tor} = 0, 10, 20, \ldots$.

- $(\lambda, \epsilon) \to (k, e)$ (velocity-space): A transformation property under integration of velocity-space integrals over θ is used to recast the velocity-space integration. Then, in both ϵ and λ, an exact Gauss-Legendre quadrature scheme is numerically generated (by nonlinear root-finding) at run-time. This is different at each radius and for different plasma equilibria.

- **nonlinearity:** The nonlinear Poisson bracket is evaluated with a *conservative* difference-spectral analogue of the Arakawa method. This scheme ensures exact conservation of density and generalized entropy at vanishing time step (independent of grid resolution).

- **collisions:** Collisions are represented by a second-order diffusive-type operator in λ. This operator is split from the collisionless problem and a irregular-grid generalization of the Crank-Nicholson method is used.

- **time-advance:** A 2nd-order IMEX RK scheme is used, with the electron parallel motion ($\partial/\partial\theta$) treated implicitly. This is exceptionally complicated due to the use of a τ-grid, as well as the presence of the fields in the advection. However, the implicitness is crucial for the elimination of a numerical instability connected with pathological *electrostatic Alfvén waves*.

0.10.3 Software Implementation

Grid indexing

Eulerian schemes for solving the GKM equations evolve the gyro-center distribution function $f(r, \tau, n_{\rm tor}, \lambda, E)$, where r is the plasma minor radius, τ is the orbit time (a parametrization of the poloidal angle), $n_{\rm tor}$ is the toroidal mode number (a linear quantum number), λ is the cosine of the pitch angle and E is the energy. Upon discretization of all differential and integral operators, we solve difference equations for the quantities $f(i, j, n, k, e)$, with

$$i = 1, 2, \ldots, N_i \tag{53}$$
$$j = 1, 2, \ldots, N_j \tag{54}$$
$$n = 1, 2, \ldots, N_n \tag{55}$$
$$k = 1, 2, \ldots, N_k \tag{56}$$
$$e = 1, 2, \ldots, N_e \tag{57}$$

Note that in general there is also a species index, but in the present work we do not use it for data distribution.

While current supercomputers offer the promise of multi-Tflop/s performance, the newest and most powerful of these are based on a distributed-memory architecture and require efficient data distribution schemes to achieve

good parallel performance. The prototype object to be distributed in memory evenly across the entire processor space is the function $f(i, j, n, k, e)$. The distribution of an index across processors is incompatible, however, with the evaluation of operators on that index. For example, a derivative in r requires that all i should be on a processor. Even more complicated is the requirement that some operators require that more than one index be simultaneously on-processor. For example, the nonlinear convolution requires both i and n to be on-processor. A summary of distribution requirements is given in Table 11.

Table 11: Distribution requirements for different code stages (i.e., evaluation of different operators).

Stage	**On-processor indices**
Linear with field solve	i, j
Pitch-angle scattering	j, k
Energy diffusion	e
Nonlinear	i, n

The obvious tactic is to change the distribution scheme to evaluate different operators. We will describe a algorithm with uses not only the global communicator `MPI_COMM_WORLD` for the totality of processors, but two new communicators, `COMM1` and `COMM2` which define processor rows and columns.

The base distribution scheme is that used for the *linear step with field solve*

$$\textbf{BASE:} \qquad f([n], \{e, k\}, i, j) \tag{58}$$

Fig. 6 illustrates this distribution strategy in the case of 16 processors – with 4 processors in each of the 4 `COMM1` subgroups, and 4 processors in each of the 4 `COMM2` subgroups. Here, `COMM1` links all columns of a given row (horizontal arrows), and `COMM2` links rows of a given column (vertical arrows). The curly braces indicate a one-dimensional array of *stacked* indices

$$\{e, k\}_p = p \quad \text{with} \quad p = 1, 2, \ldots, N_e\, N_k \tag{59}$$

where

$$(e-1) = \frac{p-1}{N_k} \quad \text{and} \quad k - 1 = (p-1) \mod N_k \tag{60}$$

The indices $\{e, k\}$ are distributed along processor columns, the $[n]$ are distributed along rows, and i and j are stored on-processor. Note that in the general case, many stacked $\{e, k\}$ indices will appear in each column. For the 16-processor case shown in Fig. 6, we would have $\{e, k\}_p$ for $p = 1, 5, 9, \ldots$ in column 1, and so on. With regard to columns, however, we make an important **simplifying assumption** and consider the number of rows to be exactly equal to N_n. This restriction suits our immediate purposes well enough, and generalization to more than one n per row is reasonably straightforward.

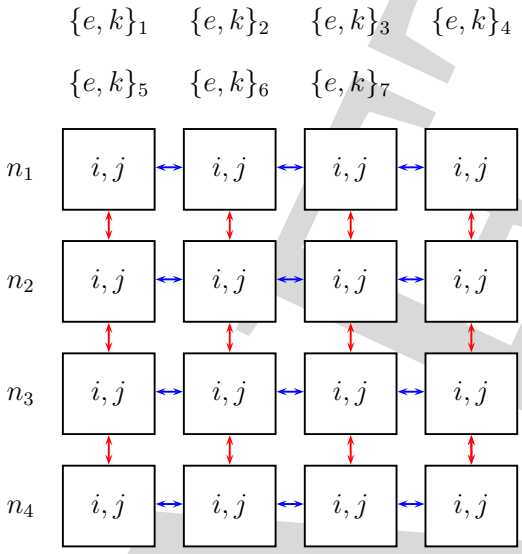

Figure 6: Base grid distribution scheme, with all i, j on processor, n distributed along rows, and e, k distributed along columns. Data redistribution will occur only only along rows or columns, thus limiting the all-to-all exchange size.

0.11 Metric Problem

0.11.1 Intent

In tokamak plasmas, performance is limited by turbulent radial transport of both energy and particles driven by so-called drift-wave instabilities. These instabilities, which are driven by gradients in the plasma temperature and density, are unavoidable and will persist in reactor-scale devices. The bulk of this transport arises from ion-scale instabilities, with time and space scales comparable to the a/v_i and ρ_i respectively (see Table 12). These relatively large-scale, slow instabilities, driven by ion-temperature-gradient (ITG) and trapped-electron (TE) modes, can nevertheless coexist with small-scale, fast instabilities driven by electron-temperature-gradient (ETG) modes. In terms of the key parameter $\mu = \sqrt{m_i/m_e}$, which is approximately 60 in a pure-deuterium plasma, the electron space scales are smaller than the ion scales by a factor of μ, while the corresponding time scales are shorter by a factor of μ. Specifically, $\rho_i = \mu\rho_e$, $k_\theta\rho_i = \mu k_\theta\rho_e$ and $a/v_i = \mu(a/v_e)$. In the past, transport from ETG instabilities was generally thought to be limited by this small parameter because the naïve electron-scale diffusivity (see Table 12) is a factor of μ smaller than the ion-scale diffusivity:

$$\frac{\chi_{\mathrm{GB}e}}{\chi_{\mathrm{GB}i}} = \frac{1}{\mu} \sim \frac{1}{60} \,. \tag{61}$$

However, landmark multi-scale simulations [**?**] which carefully treated the coupling of electron-scale to ion-scale turbulence uncovered a number of new

Table 12: List of key physical parameters.

Quantity	Definition	ITER value at mid-radius
a	Plasma minor radius	200 cm
ρ_i	Ion gyroradius	0.3 cm
v_i	Ion sound speed $\sqrt{T_e/m_i}$	8×10^5 m/s
$\chi_{\mathrm{GB}i}$	Ion gyroBohm diffusivity $\rho_i^2 v_i/a$	4 m^2/s
ρ_e	Electron gyroradius	0.005 cm
v_e	Electron velocity $\sqrt{T_e/m_e}$	5×10^7 m/s
$\chi_{\mathrm{GB}e}$	Electron gyroBohm diffusivity $\rho_e^2 v_e/a$	0.07 m^2/s

and provocative results

1. the simplified ETG-ai model (i.e., ETG with adiabatic ion dynamics) gives physically nonsensical results for some parameters,
2. adding additional long-wavelength physics, such as equilibrium $\mathbf{E} \times \mathbf{B}$ shear or nonadiabatic ion dynamics, to the ETG-ai model tends to restore physically sensible behavior
3. the back-reaction of ETG on ITG turbulence is insignificant (i.e., small scale effects do not affect large-scale structures)
4. short-wavelength ETG transport is reduced as long-wavelength ITG turbulence increases in strength,
5. in the absence of $\mathbf{E} \times \mathbf{B}$ shear stabilization, most of the electron energy transport arises from ion scales ($k_\theta \rho_i < 1.0$).

However, these studies were carried out with an increased electron mass $\mu = 30$ in order to make the problem, which has a cost-scaling of approximately $\mu^{3.5}$, computationally feasible. The expectation is that while the absolute transport levels, in particular at scales smaller, will depend on μ, the qualitative features of the problem will be preserved. Comparisons between $\mu = 20$ and $\mu = 30$ confirmed this expectation. However, to obtain physically accurate transport levels, simulations with the true mass ratio are required.

0.11.2 Initial and Boundary Conditions

For these benchmarks, where possible, the nominal local parameters are derived from the Cyclone base-case [**?**]; namely, we use a simple unshifted-circle equilibrium with $q = 1.4$, $s = 0.786$, $r/a = 0.5$, $R/a = 2.775$, $a/L_{Te} = 2.484$, $a/L_{ne} = 0.8$, $n_e = n_i$ and $T_e = T_i$. Unlike the original work of Jenko and Dorland, particle trapping (finite-r/R) is retained.

Furthermore, this case uses a 128-point velocity-space grid (8 energies, 8 pitch angles and two signs of velocity), and the parallel (to the field line) resolution is taken to be 10 points per passing particle orbit. These choices have been carefully studied and justified in previous work [**?**]. The details of the perpendicular resolution, characterized by $L_x \doteq 2\pi/(k_r)_{\min}$ and $L_y \doteq 2\pi/(k_\theta)_{\min}$, are

given in each case. We used a 4th-order explicit Runge-Kutta time-integration scheme, with a nominal simulation timestep of $(v_e/a)\Delta t = 0.03$ (with minor variation in some cases). Here, "min" refers to the minimum nonzero value of the respective wavenumber. All simulations use radially periodic (flux-tube) boundary conditions.

0.11.3 Application Metric

The observable that GYRO code developers use to relate performance of their application to the performance of a computer chosen to execute their application is the *number of timesteps per second per process.* This is due to the fact that typically GYRO simulations run for 10^4 to 10^5 timesteps. So the timesteps-per-second metric is the metric that best represents the computational cost of a simulation.

0.12 Q2 Metric Status

The key parameter for the Q2 benchmark is the mass ratio, $\mu = 30$. As previously discussed, this parameter has significant computational cost-scaling associated with it: $\mu^{3.5}$. For example, the cost increase in going from $\mu = 30$ to $\mu = 40$ (at fixed domain size and velocity-space resolution) is $(40/30)^{3.5} \simeq 2.7$.

0.12.1 Results

The two baseline runs were run again on 4608 cores (processes) on the quad-core Jaguar system using 4 MPI processes on a socket (each core of 1152 sockets had an MPI process). The default programming environment on Jaguar had changed significantly since the time the original runs were performed and thus the two runs were performed again. Rerunning the Q2 benchmark allows us to make sure the programming environment and performance analyses are consistent for the Q2 and Q4 problems.

The following table shows the new operation counts that are averages over processes obtained from Cray's CrayPat performance tool[7].

[7]The April 2008 data is still available if necessary.

20 timesteps		
Time%		100.0%
Time		60.009599
PAPI_L1_DCM	10.051M/sec	688919532 misses
PAPI_TOT_INS	2813.323M/sec	192827045298 instr
PAPI_L1_DCA	1062.503M/sec	72824635403 refs
PAPI_FP_OPS	908.743M/sec	62285846656 ops
User time (approx)	68.541 secs	1.43935E+11 cycles
Cycles	68.541 secs	1.43935E+11 cycles
User time (approx)	68.541 secs	1.43935E+11 cycles
Utilization rate		99.3%
Instr per cycle		1.34 inst/cycle
HW FP Ops / Cycles		0.43 ops/cycle
HW FP Ops / User time	908.743M/sec	62285846656 ops
HW FP Ops / WCT	902.568M/sec	
HW FP Ops / Inst		32.3%
Computation intensity		0.86 ops/ref
MIPS	12963793.36M/sec	
MFLOPS	4187487.52M/sec	
10 timesteps		
Time%		100.0%
Time		51.783673
PAPI_L1_DCM	10.812M/sec	556355021 misses
PAPI_TOT_INS	2802.185M/sec	144194070016 instr
PAPI_L1_DCA	1068.526M/sec	54983892499 refs
PAPI_FP_OPS	922.300M/sec	47459434572 ops
User time (approx)	51.458 secs	1.08061E+11 cycles
Cycles	51.458 secs	1.08061E+11 cycles
User time (approx)	51.458 secs	1.08061E+11 cycles
Utilization rate		99.4%
Instr per cycle		1.33 inst/cycle
HW FP Ops / Cycles		0.44 ops/cycle
HW FP Ops / User time	922.300M/sec	47459434572 ops
HW FP Ops / WCT	916.494M/sec	
HW FP Ops / Inst		32.9%
Computation intensity		0.86 ops/ref
MIPS	12912468.93M/sec	
MFLOPS	4249956.15M/sec	

By measuring the instruction count for both 10 and 20 timesteps, we can take the difference to remove the startup cost and thus determine the cost of a single timestep. Therefore the PAPI_TOT_INS for a single timestep for one process is (192827045298-144194070016)/10 = 4863297528. The time for one timestep is (69.01-51.78)/10=1.723[8], so the instruction rate is 4863297528/1.723 = 2822.58M/sec.

From the point-of-view of users, the timestepping performance is by far the greatest concern since a typical simulation will make 10^4 to 10^5 timesteps as mentioned above. The internal timers of GYRO report that a timestep takes 1.54 seconds (in contrast to the CrayPAT data showing 1.723 seconds), or in

[8]This assume both tests ran at exactly same rate.

terms of the application metric 0.6481 timesteps per second. Of course, the larger this number is, the better[9].

0.13 Q4 Metric Status

The key parameter for the Q4 benchmark problem is $\mu = 40$. The Q4 benchmark mass ratio is closer to the physically correct value of $\mu = 60$. The Q4 benchmark also has a larger domain size, which is necessary to properly treat the long-wavelength (ion-scale) modes which typically dominate the transport physics. Both these changes increase the level of physical realism, and should give fluxes which are closer to those observed in experiments. The energy grid was also increased to ensure that we are fully resolved in velocity space. A detailed list of differences in the input parameters for the Q2 and Q4 cases is shown in the table below:

INPUT Parameter	Q2	Q4	Comment
RADIAL_GRID	375	1000	Radial gridpoints
PASS_GRID	4	8	Velocity gridpoints (passing particles)
TRAP_GRID	4	8	Velocity gridpoints (trapped particles)
ENERGY_GRID	8	32	Velocity gridpoints (energies)
TOROIDAL_GRID	72	192	Toroidal Fourier harmonics
RADIAL_GYRO_BAND	61	81	Gyroaverage stencil width
TOROIDAL_SEP	60	30	Inverse of binormal domain size
BOX_MULTIPLIER	49.0	98.0	Radial domain size
TIME_STEP	0.0012	0.000875	Time step
MU_2	30.0	40.0	Root of the mass ratio

As a bottom-line measure of the size of the two cases, we can simply enumerate the total number of gridpoints at which the distribution function is discretized. These totals are 69M for Q2 and 3.9B for Q4, giving a ratio of 57. Thus, as a lower bound, we estimate that there is *at least* 57 times more work for Q4 than for Q2.

0.13.1 Results

Again two runs were performed, this time on 24576 cores (processes) on the quad-core Jaguar system using 4 MPI processes on a socket (each core of 6144 sockets had an MPI process). The following table shows operation counts that are averages over processes obtained from Cray's CrayPat performance tool.

[9] This was 0.47537 with the April 2008 programming environment.

20 timesteps		
Time%		100.0%
Time		1248.581875
PAPI_L1_DCM	25.398M/sec	31672389924 misses
PAPI_TOT_INS	1816.819M/sec	2265634132620 instr
PAPI_L1_DCA	636.660M/sec	793936308596 refs
PAPI_FP_OPS	647.919M/sec	807977265171 ops
User time (approx)	1247.034 secs	2618770711634 cycles
Cycles	1247.034 secs	2618770711634 cycles
User time (approx)	1247.034 secs	2618770711634 cycles
Utilization rate		99.9%
Instr per cycle		0.87 inst/cycle
HW FP Ops / Cycles		0.31 ops/cycle
HW FP Ops / User time	647.919M/sec	807977265171 ops
HW FP Ops / WCT	647.116M/sec	
HW FP Ops / Inst		35.7%
Computation intensity		1.02 ops/ref
MIPS	44650137.11M/sec	
MFLOPS	15923266.32M/sec	
10 timesteps		
Time%		100.0%
Time		1095.823138
PAPI_L1_DCM	27.408M/sec	30002445406 misses
PAPI_TOT_INS	1616.131M/sec	1769111957526 instr
PAPI_L1_DCA	577.419M/sec	632076505704 refs
PAPI_FP_OPS	511.801M/sec	560247269725 ops
User time (approx)	1094.659 secs	2298782879229 cycles
Cycles	1094.659 secs	2298782879229 cycles
User time (approx)	1094.659 secs	2298782879229 cycles
Utilization rate		99.9%
Instr per cycle		0.77 inst/cycle
HW FP Ops / Cycles		0.24 ops/cycle
HW FP Ops / User time	511.801M/sec	560247269725 ops
HW FP Ops / WCT	511.257M/sec	
HW FP Ops / Inst		31.7%
Computation intensity		0.89 ops/ref
MIPS	39718044.41M/sec	
MFLOPS	12578020.20M/sec	

By measuring the instruction count for both 10 and 20 timesteps, we can take the difference to remove the startup cost and thus determine the cost of a single timestep. Therefore the PAPI_TOT_INS for a single timestep for one process is (2265634132620-1769111957526)/10 = 49652217509. The time for one timestep is (1248.58-1095.82)/10 = 15.276[10] So the instruction rate is 49652217509/15.276 = 3250.34M/sec.

The internal timers of GYRO report that a timestep takes approximately 17.1 seconds (in contrast to the CrayPAT data showing 15.276 seconds), or in terms of the application metric 0.0585 timesteps per second.

[10] This assume both tests ran at exactly same rate.

Table 13: GYRO Metric Benchmark Summary

	Q4	Q2	ratio
# cores	24576	4608	5.333
μ	40	30	2.053
time	152.75	17.22	8.868
TOT_INS	4.965E+11	4.863E+10	10.209
FP_OPS	2.477E+11	1.482E+10	16.709

0.14 Metric Interpretation

In this section, we analyze the benchmark data presented in the preceding sections. Table 13 shows a summary of the results for performing 10 timesteps.

In a nutshell, the data shows that for 10 timesteps the Q4 benchmark used 5.3 times more processors to do 10.2 times more instructions per processor and 16.7 times more operations per processor while only taking 8.9 times longer. The total work (54 times the total number of instructions and 89 times the number of operations) is roughly consistent with the factor of 57 increase in total gridpoint count previously noted. The numbers (54,89,57) – all measures of the increase in problem size – are to be compared with the total CPU time, which increased by a factor of 47. Thus, by all measures, GYRO achieved a slightly better-than-linear scaling with problem size.

Computational Science Capability: PFLOTRAN

0.15 Introduction

Over the past several decades, subsurface (groundwater) flow and transport models have become vital tools for the U.S. Department of Energy (DOE) in its environmental stewardship mission. These models have been employed to evaluate the impact of alternative energy sources and the efficacy of proposed remediation strategies for legacy waste sites. For years, groundwater models remained arguably simple in comparison to the sophisticated models of today. These traditional models simulated single-phase groundwater flow and single-component solute transport within a single continuum, with basic chemical reactions such as radioactive decay and linear sorption to rock/soil surfaces. Although these simplified groundwater models are still widely used today, advances in subsurface science have enabled the development of more sophisticated models that employ multiple fluid phases and chemical components coupled through a suite of biological and geochemical reactions at multiple scales. However, with this increased complexity comes the need for more computational power, typically beyond that of the average desktop computer. This is especially true when applying these sophisticated multiphase flow and multicomponent reactive transport models to two- or three-dimensional problem domains.

0.16 Background and Motivation

As part of the SciDAC groundwater science application area, PFLOTRAN is being utilized to simulate radionuclide transport at the U.S. Department of Energy's Hanford Site in southeastern Washington State. Throughout the Cold War, Hanford served as a facility within the DOE nuclear weapons complex for the production of plutonium for national defense. Plutonium production initiated during the latter years of World War II and continued until the 1980s, after which efforts at the Hanford Site shifted toward environmental remediation.

Of particular interest to DOE is the cleanup/remediation of the Hanford 300 Area, which is located in the southeast corner of the Hanford Site, immediately north of the city of Richland along the Columbia River. During the

Cold War, facilities at the Hanford 300 Area were utilized to research and fabricate nuclear fuels for plutonium production at the Hanford Site. As a result of this fuel production, liquid and solid wastes were often disposed of in process ponds and process trenches within the confines of the Hanford 300 Area. This waste is known to have contained uranium, though the exact amount released is uncertain. Over time, this waste has migrated downward to the water table polluting the underlying groundwater.

Significant simulation and modeling efforts were put forth in the early 1990s in an attempt to determine the rate at which uranium within the groundwater would migrate toward and discharge into the Columbia River. Unfortunately, these models lacked the sophistication necessary to accurately simulate uranium transport, predicting that the uranium would be removed or flushed out of the Hanford 300 Area subsurface by ambient groundwater flow within a decade. Today, nearly 15 years later, uranium concentrations at the site remain relatively unchanged. Several possibilities exist attempting to explain the discrepancies between the predicted and observed transport of uranium. These include: (1) fluctuations in the stage of the Columbia river and its impact on groundwater velocities, (2) the vadose zone contribution to the source term, (3) inadequacies in the uranium transport conceptual model due to overly simplistic reactions representing the sorption of uranium on sediment grains; and, (4) uncertainty of the uranium source term mass concentrations and leaching rates.

The conceptual model for flow at the Hanford 300 Area is complicated by the highly permeable Hanford Unit and the rapidly fluctuating Columbia River producing changes in magnitude and direction of flow. The soils beneath the Hanford 300 Area are composed of layers of high and low permeability soil, the top layer of which is the highly permeable Hanford Unit located near the water table. The Hanford Unit is composed of highly permeable cobbles, gravels, and sands. Hydraulic conductivities within the Hanford Unit are on the order of a thousand to tens of thousands of meters per day. In comparison, the Ringold Units below the Hanford Unit exhibit much lower hydraulic conductivities of 0.01 to 150 meters per day. Thus, within the Hanford Unit, where all original uranium sources are believed to have resided, groundwater has the potential of flowing rapidly with very small pressure gradients in the aquifer.

Rapid fluctuations in river stage, as shown in Figure 7, can produce these pressure gradients. Diurnal (daily) fluctuation in river stage can be up to 1.5 meters, while seasonal fluctuation can exceed 2.5 meters. The variable release of water from Priest Rapids Dam upriver is the cause of the diurnal fluctuation, while the seasonal variation is due to snow melt, irrigation, etc. This fluctuation in river stage along with the highly permeable Hanford Unit directly impacts the water table beneath the Hanford 300 Area, causing river waters to flow into the area during high stage and retreat during low stage. One could view this phenomenon as a slow tidal effect where sources of uranium residing immediately above the water table in the vadose zone are cyclicly flushed over time as the water table rises and falls, introducing new uranium mass into the underlying aquifer. An accurate depiction of the flow processes beneath the Hanford 300 Area is critical for capturing such behavior in the uranium transport model.

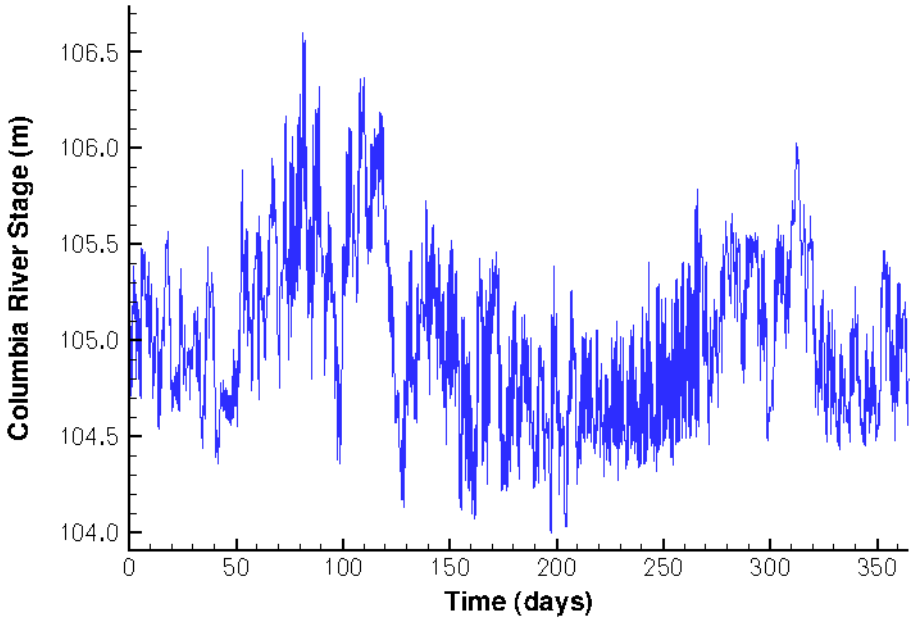

Figure 7: Fluctuation in the Columbia river stage.

0.17 Capability Overview

PFLOTRAN is a parallel multiphase flow and multicomponent reactive (geochemical) transport code that originated from the serial FLOTRAN code developed at Los Alamos National Laboratory. The code is composed of different modules for flow and transport, with the option of running the modules in coupled or decoupled mode. PFLOTRAN is capable of simulating fluid flow though porous media with the following fluid phases: air, water, supercritical CO_2. PFLOTRAN-generated fluid flow velocities or fluxes are utilized by the transport module to compute solute transport. Within PFLOTRAN, transport and reaction are fully coupled. PFLOTRAN's problem domain is discretized spatially using the integrated finite volume approach, with fully-implicit backward-Euler time differencing.

PFLOTRAN's parallel paradigm is based on domain decomposition where the computational problem domain is divided into subdomains, one domain assigned to each processor employed. PFLOTRAN then leverages PETSc solvers and data structures to link these subdomains within a single parallel code. Since PFLOTRAN is founded upon these PETSc data structures, the code is capable of utilizing the full suite of algorithms available in the library. In addition, PETSc also provides linkage to a variety of external software packages (e.g. Hypre, Trilinos, Zoltan). Within PFLOTRAN, the distributed array (DA) is the key PETSc data structure that dictates the layout and decomposition of the three-dimensional parallel domain. From the DA, PETSc generates parallel matrices (Mat) and vectors (Vec) with consistently mapped local and global indexing for the parallel communication required to solve the problem. Therefore, although an indepth understanding of parallel communication paradigms such as the message passing interface (MPI) is helpful, it is not required since PETSc hides the communication from the end user. For example, when employing the Newton-Raphson method to solve a nonlinear system of equations in parallel using the PETSc nonlinear solver or SNES, the programmer essentially provides functions to PETSc that evaluate the residual and compute the Jacobian for the nodes that reside locally on each processor. PETSc then utilizes these functions to compute a solution vector, which is then returned to PFLOTRAN for boundary condition updates, I/O, etc. Thus, the application scientist is able to focus more on the science (in this case, subsurface physics and chemistry) rather than solvers, preconditioners, etc.

0.17.1 Physical Model

The physical model used in the benchmark study is based on Richards equation for fluid flow in a partially saturated medium. In this approximate formulation the gas phase is assumed to be inert. Richards equation is coupled to the advection-diffusion-reaction equation for solute transport. Richards equation may be represented in the form

$$\frac{\partial}{\partial t}(\varphi s \rho) + \boldsymbol{\nabla} \cdot \rho \boldsymbol{u} \;=\; \mathcal{S}, \tag{62}$$

where φ denotes the porosity of the porous with saturation s, ρ denotes the fluid density, $\mathcal{S}$ refers to a source/sink term, and $\boldsymbol{u}$ denotes the Darcy velocity

defined by

$$\boldsymbol{u} \;=\; -\frac{\kappa\kappa_r}{\mu}\boldsymbol{\nabla}\big(P-\rho g z\big), \tag{63}$$

with fluid pressure P, viscosity μ, vertical distance z, saturated permeability κ, relative permeability κ_r, and acceleration of gravity g. The relative permeability is a function of saturation for which the van Genuchten expression

$$\kappa_r(s) \;=\; \sqrt{s_{\text{eff}}}\left\{1-\left[1-\left(s_{\text{eff}}\right)^{1/\lambda}\right]^{\lambda}\right\}^2, \tag{64}$$

is used where s_{eff} is defined by

$$s_{\text{eff}} \;=\; \frac{s-s_r}{1-s_r}, \tag{65}$$

where s_r denotes the residual saturation. The capillary pressure P_c is related to saturation by the equation

$$s_{\text{eff}} \;=\; \left[1+\left(\alpha|P_c|\right)^m\right]^{-\lambda}, \tag{66}$$

where the quantities m and λ are related by the expressions

$$\lambda \;=\; 1-\frac{1}{m}, \qquad m \;=\; \frac{1}{1-\lambda}. \tag{67}$$

Fluid and capillary pressures are related to the gas pressure, which in this case is taken as constant, by

$$P \;=\; P_g - P_c. \tag{68}$$

Note that κ_r = 0 for saturations below the residual saturation.

The solute transport equation for a single component reactive species has the form

$$\frac{\partial}{\partial t}\big(\varphi s C\big) + \boldsymbol{\nabla}\cdot\big(\boldsymbol{u}C - \varphi s D\boldsymbol{\nabla}C\big) \;=\; \mathcal{S}_C, \tag{69}$$

for concentration C, diffusion/dispersion coefficient D, and source/sink term $\mathcal{S}_C$. The source term is represented by a first-order linear reaction rate given by

$$\mathcal{S}_C \;=\; -k\big(C-C_{\text{eq}}\big), \tag{70}$$

with an effective kinetic rate constant k and equilibrium concentration C_{eq}. The transport equation is coupled to the flow equation (62) through the saturation s and Darcy flow velocity $\boldsymbol{u}$.

0.17.2 Numerical Method

A standard finite volume approach is used to discretize the governing equations with a fully implicit backward Euler time stepping method. For Richards equation given in Eqn.(62), discrete finite volume equations combined with the above constitutive relations are derived below.

Partitioning the computational domain into a set of finite volumes V_n and integrating the partial differential equations over each volume yields a discretized form of the mass conservation equations. The following results are obtained for the accumulation, source, and flux terms:

$$\int_{V_n} \frac{\partial}{\partial t}(\varphi s\rho)\, dV \simeq \varphi \frac{(s\rho)_n^{t+\Delta t} - (s\rho)_n^t}{\Delta t} V_n, \tag{71}$$

for the accumulation term,

$$\int_{V_n} \mathcal{S}\, dV \simeq \mathcal{S}_n V_n, \tag{72}$$

for the source term, and

$$\int_{V_n} \boldsymbol{\nabla} \cdot \rho \boldsymbol{u}\, dV \;=\; \int_{\partial V_n} \rho \boldsymbol{u} \cdot \boldsymbol{dS} \;=\; \sum_{n'} (\rho u)_{nn'} A_{nn'}, \tag{73}$$

for the flux term, where ∂V_n denotes the surface of V_n, and the sum is over the neighboring volumes connected to V_n with the flux $(\rho \boldsymbol{u})_{nn'}$ across the $n-n'$ interface connecting volumes V_n and $V_{n'}$. The subscript nn' indicates that the quantity is evaluated at the interface, the quantities d_n, $d_{n'}$ denote the distances from the centers of the control volumes V_n, $V_{n'}$ to the their common interface with interfacial area $A_{nn'}$. The term involving the Darcy flux is further discretized as

$$(q\rho)_{nn'} \;=\; -\left(\frac{kk_r}{\mu}\right)_{nn'} \left(\frac{P_n - P_{n'}}{d_n + d_{n'}} - (W\rho)_{nn'} g \cos\theta_{nn'}\right), \tag{74}$$

where W is the average molecular weight. $\theta_{nn'}$ represents the angle between the line drawn between node $n-n'$ and the direction of gravity. Combining these results gives the residual equation for the discretized form of the partial differential equations

$$R_n \;=\; \varphi_n \big((s\rho)_n^{k+1} - (s\rho)_n^k\big) \frac{V_n}{\Delta t} + \sum_{n'} (\rho u)_{nn'}^{k+1} A_{nn'} - \mathcal{S}_n V_n, \tag{75}$$

where, in general, R_n is a nonlinear function of the independent field variables.

A similar procedure is followed for the tracer transport equation Eqn.(69). Upstream weighting is used to compute the advective fluxes across an iterface. The residual equation has the form

$$R_n \;=\; \varphi_n \big((sC)_n^k - (sC)_n^{k+1}\big) \frac{V_n}{\Delta t} + \sum_{n'} \left[(uC^{k+1})_{nn'} - \varphi D_{nn'} \frac{C_{n+1}^{k+1} - C_n^{k+1}}{d_{n+1} + d_n} \right] A_{nn'} - \mathcal{S}_n V_n, \tag{76}$$

with diffusivity $D_{nn'}$. The interface concentration $C_{nn'}$ is computed from the equation

$$C_{nn'} \;=\; \begin{cases} C_n, & u_{nn'} > 0, \\ C_{n'}, & u_{nn'} < 0, \end{cases} \tag{77}$$

where $u_{nn'}$ is taken as positive for pressure and concentration, respectively, for flow into the control volume V_n.

These equations are solved using a Newton-Raphson iteration technique in which the discretized equations are first linearized resulting in the Newton-Raphson equations

$$\sum_{n'} J^k_{nn'}\, \delta x^{k+1}_{n'} \;=\; -R^k_n, \tag{78}$$

for unknowns δx^k_n at the kth iterate, with the Jacobian matrix $J_{nn'}$ defined by

$$J_{nn'} \;=\; \frac{\partial R_n}{\partial x_{n'}}. \tag{79}$$

The Jacobian matrix is evaluated numerically using a finite difference method. This approach gives PFLOTRAN greater flexibility in applications to various systems, without any change or only minor modification to the EoS module.

For reactive transport it proves convenient if not essential in many problems to solve for the logarithm of the concentration, rather than the concentration itself. For this case the usual linear update relation

$$x^{(k+1)}_n \;=\; x^k_n + \delta x^{(k+1)}_n, \tag{80}$$

is replaced with

$$x^{(k+1)}_n \;=\; x^{(k)}_n \mathrm{e}^{\ln \delta x^{(k+1)}_n}. \tag{81}$$

For Richards equation the residual equations are solved for pressure as the unknown variable. Although in the Richards approximation the gas phase has constant pressure and is assumed to be inert, it is necessary to test for a phase change from two-phase gas-liquid conditions to a single phase liquid and vice versa.

PFLOTRAN utilizes a modular object-oriented, Fortran 90 implementation (see Figure 8) using a finite volume discretization method combined with backward-Euler time differencing to solve the systems of equations governing subsurface flow and transport. Upwinding is used for the advective term in the transport equations. With a highly-scalable parallelization using domain decomposition through tight integration of Argonne National Laboratory's PETSc library, [**?**], PFLOTRAN runs on any computer platform supported by PETSc, which is essentially all platforms since PETSc developers willingly support any platform.

Flow Chart Definitions (see Figure 8):

1. Simulation object: Highest level data structure providing all information for running a simulation

2. Timestepper object: Pointer to Newton-Krylov solver and tolerances associated with time stepping

3. Solver object: Pointer to nonlinear Newton and linear Krylov solvers (PETSc SNES/KSP/PC) along with associated convergence criteria

4. Realization object: Pointer to all discretization and field variables associated with a single realization of a simulation

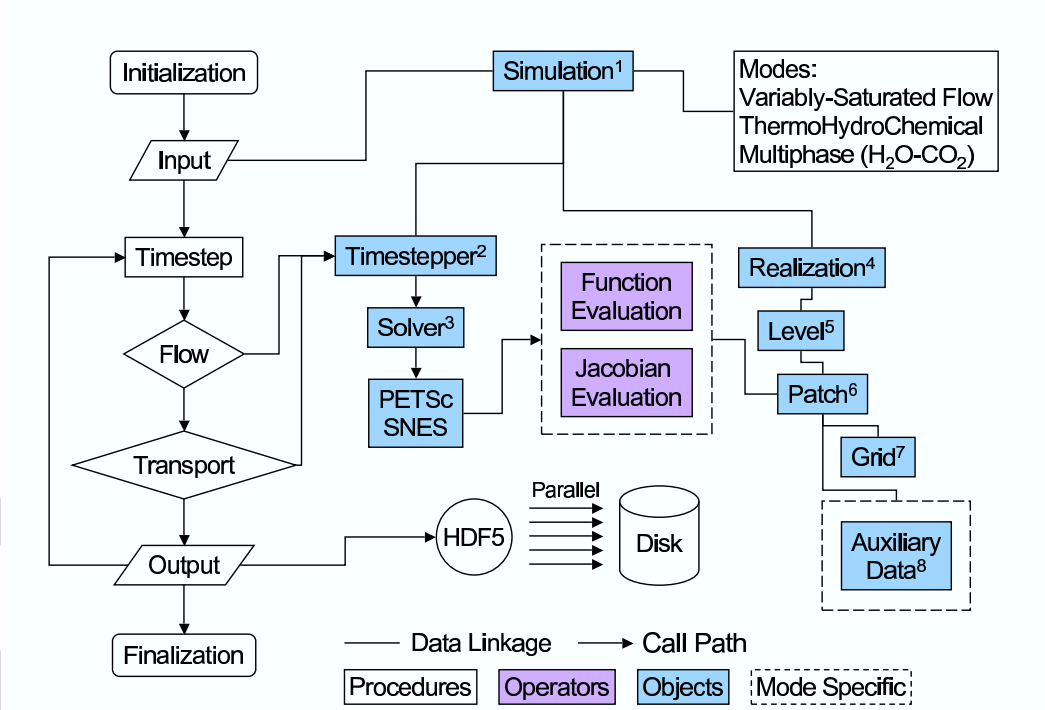

Figure 8: PFLOTRAN flow diagram illustrating use of procedures, operators, objects, and mode specific operators and objects.

5. Level object: Pointer to discretization and field variables associated with a single level of grid refinement within a realization
6. Patch object: Pointer to discretization and field variables associated with a subset of grid cells within a level
7. Grid object: Pointer to discretization within a patch
8. Auxiliary Data object: Pointer field variables within a patch

0.17.3 Software Implementation

Key features/capabilities of PFLOTRAN either implemented or currently being implemented(*) include:

- Object-oriented Fortran 90 data structures
- PETSc solvers/preconditioners
- Modular linkage to physicochemical processes
- Collective parallel I/O through HDF5
- Adaptive mesh refinement (AMR)*
- Multicontinuum subgrid model*
- Multiphase flow
- Thermal transport
- Multicomponent reactive transport
- Biogeochemistry

0.17.4 Execution Performance

The performance of PFLOTRAN is demonstrated for the Hanford 300 Area. The code is being enhanced and employed to simulate groundwater flow and U(VI) transport at this site. The real-world Hanford 300 Area problem provides a challenging computational problem to demonstrate the capabilities of the code.

Hanford 300 Area Application Conceptual Model

- The PFLOTRAN model of the Hanford 300 Area consists of a gridded domain measuring 1350 $\times$ 2500 $\times$ 20 meters (x, y, z) with orientation aligned with the Columbia River at 14° of west of north (Figure 11). The base of the model lies at 90 meters elevation above sea level.
- The three grid resolutions simulated in this work include:
  - 20 meter horizontal $(x-y)$ $\times$ 1 meter vertical (z) = 170,000 degrees of freedom (170K dof)

  - 10 meter horizontal $\times$ 0.5 meter vertical = 1,350,000 degrees of freedom (1.35M dof)
  - 5 meter horizontal $\times$ 0.25 meter vertical = 10,800,000 degrees of freedom (10.8M dof)

- Stratigraphy was mapped to each grid from the Hanford EarthVision database
  [**?**]. The two predominant geologic units simulated in the current conceptual model are the Hanford and Ringold units (the contact between which is shown in Figure 9). These units are shown near the top of Figure 10, a 60-meter deep representation of the problem domain viewed from southeast.

- Hanford 300 Area hydraulic properties were assigned based on data provided by [**?**]

- Transient hydrostatic and seepage face boundary conditions were assigned to the western inland and river boundaries, respectively, with time-varying datums and gradients. Specified surface recharges was assigned to the top of the model domain, while the north and south boundaries were no flow.

- Solute transport was simulated through an infinite rate-limited dissolution source term (normalized tracer concentration) specified near the center of the Hanford 300 Area Integrated Field-Scale Subsurface Research Challenge (IFC) site (see Figure 11). Source zone region: area = 1600 m^2 (107K dof), 2500 m^2 (1.35M, 10.8M dof); elevation = 104m–110m.

Hanford 300 Area Simulation Results

PFLOTRAN simulations run to date focus on establishing and improving the accuracy of the Hanford 300 Area flow solution and the subsequent fluxes utilized by the transport portion of the code. For these simulations, the variably-saturated Richards mode (i.e. employing Richards equation) was utilized within PFLOTRAN. Simulations were initialized to steady state (based on 10am May 1, 1992 conditions), restarted, and run transient to 7500 hours (10am May 1, 1992 to 10am March 9, 1993)

Figures 12 illustrate $x - y$ cross sections of the pressure (z = 90m) and tracer concentration (z = 105m) at 7500 hours simulation time (10am Mar. 9, 1993) for the 170K, 1.35M and 10.8M dof grids, respectively.

Figure 13 illustrates the pressure, saturation, and tracer concentration for the 10.8M dof solution viewed at 7500 hours from the southeast, with the highly-permeable Hanford unit hidden to better illustrate concentration isosurfaces. The 170K and 1.35M dof solutions are similar, though slightly more diffuse.

Figure 14 compares the piezometric head observed at monitoring well 399-3-12 (see location in Figure 11) to the PFLOTRAN heads predicted nearest to the well for each of the simulation scenarios. River stage is also provided as a reference, and is far more oscillatory than the observed head at the well. Figure 15 is an enlarged view of Figure 14 between 3250-3750 hours. All three

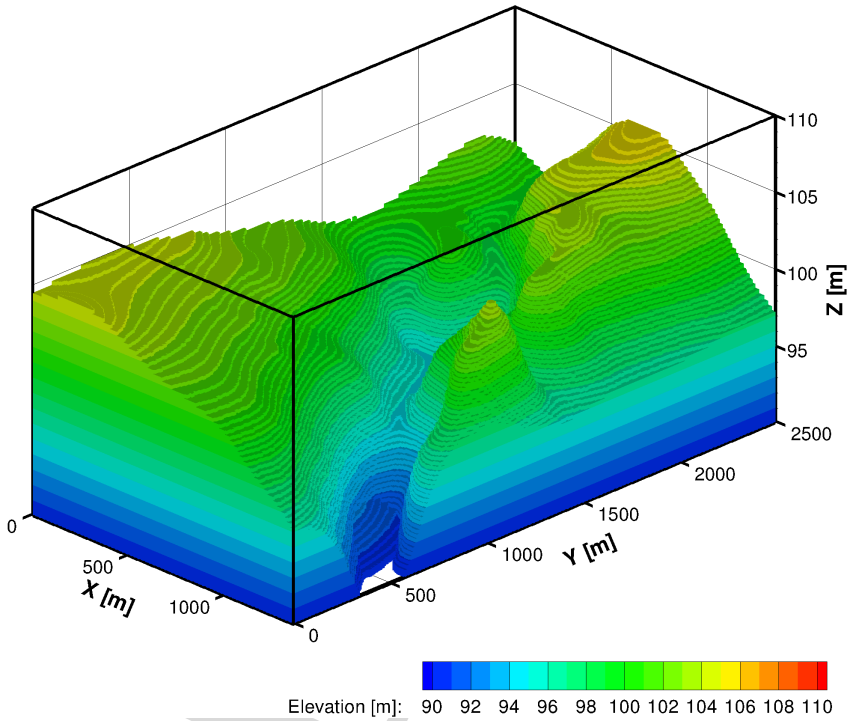

Figure 9: Top of Ringold unit (Hanford-Ringold contact). The Hanford unit is hidden.

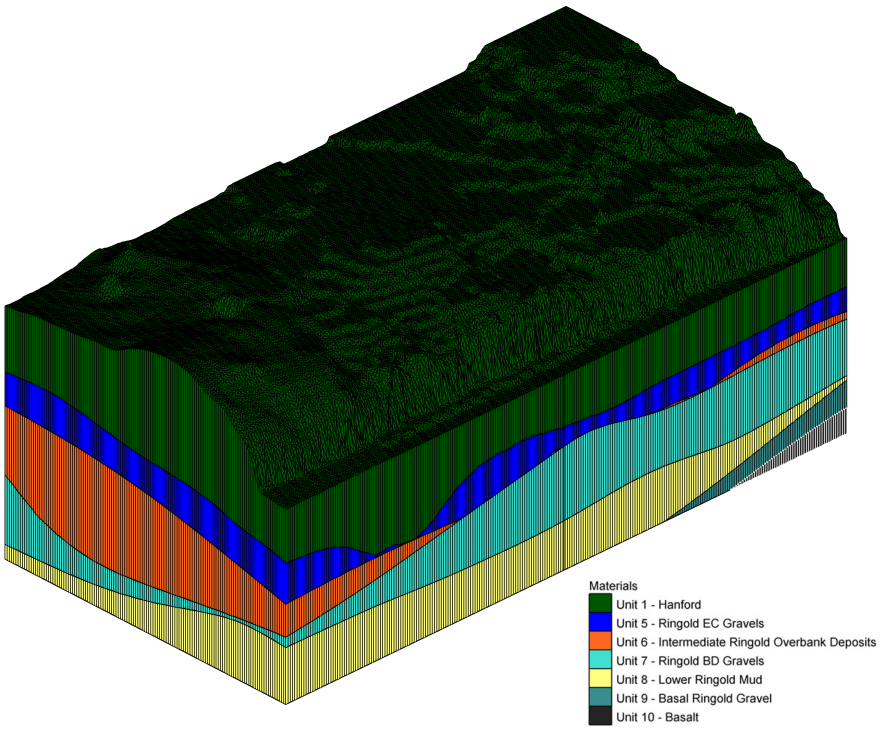

Figure 10: Hanford 300 Area stratigraphy (z scale = 20x, z axis ranges 70–130 meters).

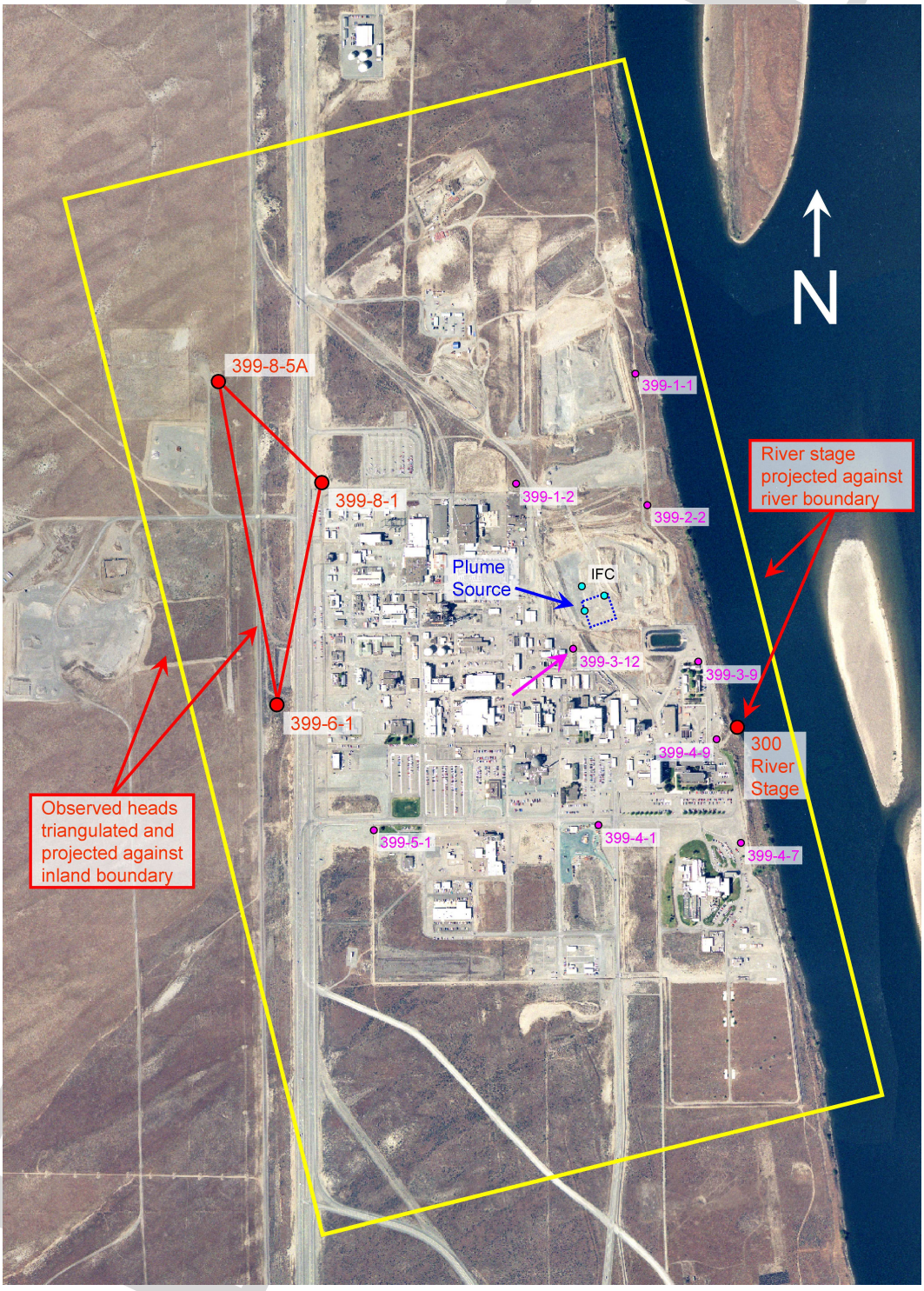

Figure 11: Layout of Hanford 300 Area showing the location of the IFC and monitoring wells.

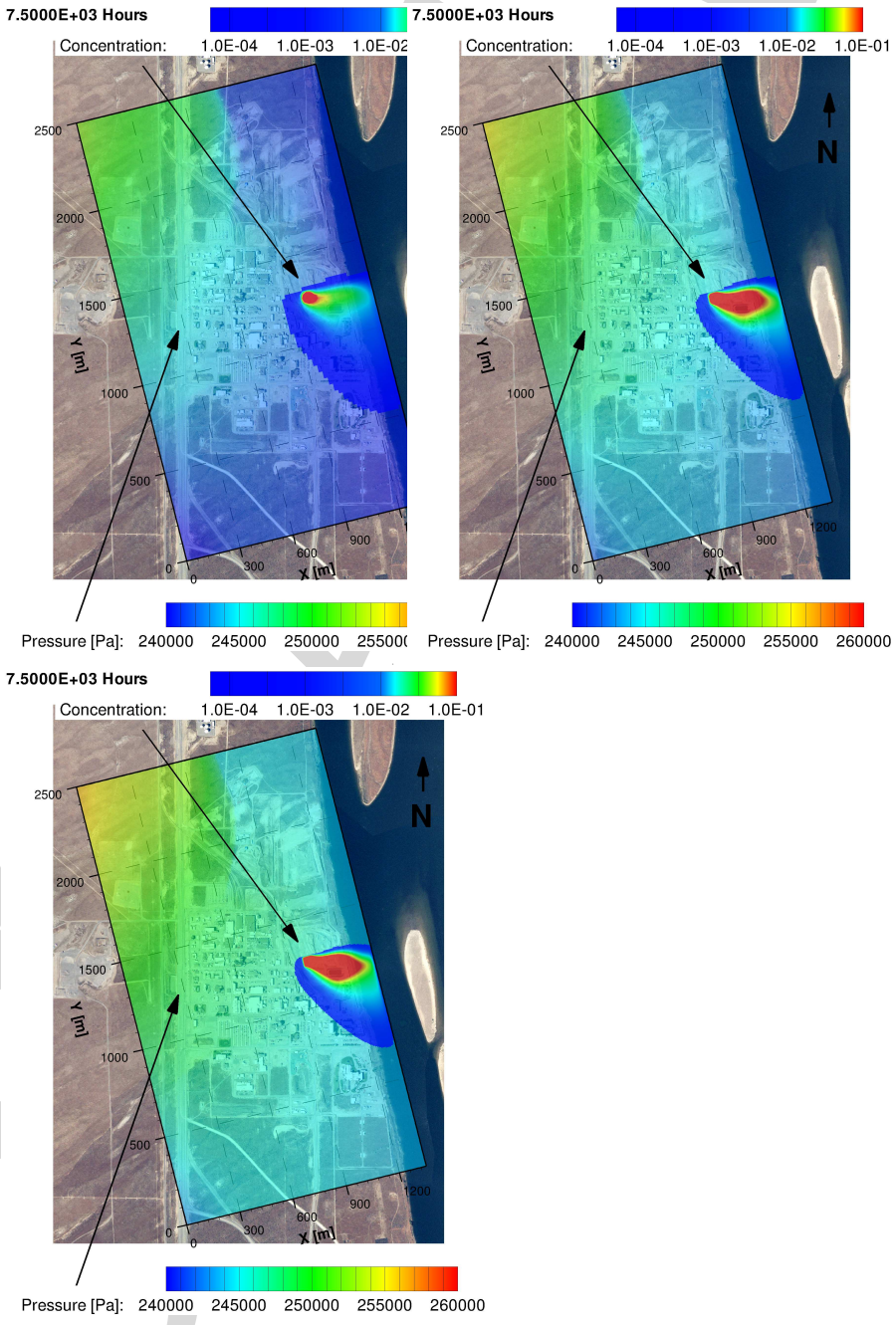

Figure 12: 170K (top left), 1.35M (top right), and 10.8M (bottom left) dof model.

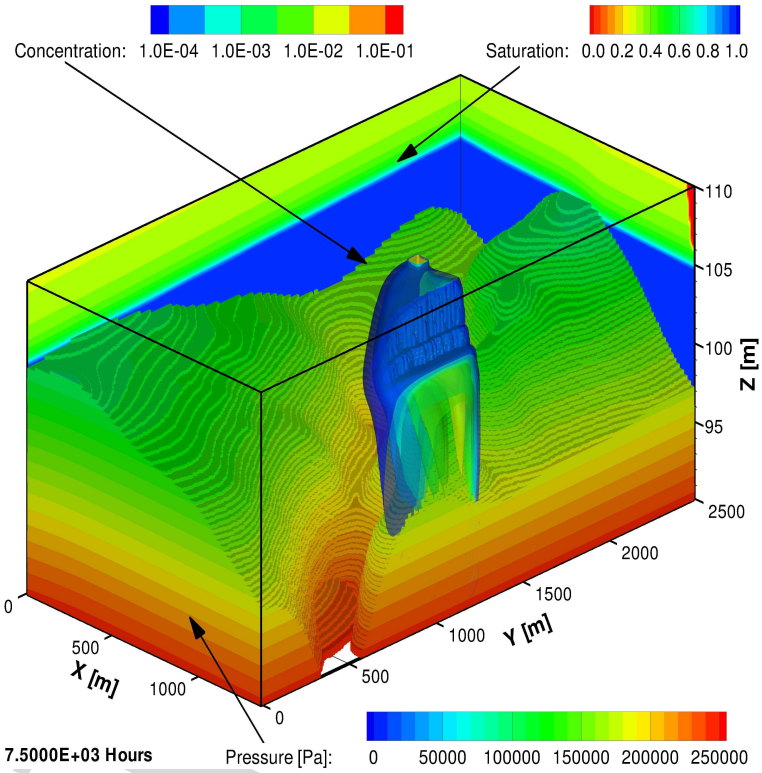

Figure 13: Pressure contours, saturation cross-sections, and tracer isosurfaces for 10.8M dof model at 7500 hours (5m $x-y$, 0.25m z resolution). The model domain is viewed from the southeast with the Hanford unit hidden to better view the isosurfaces.

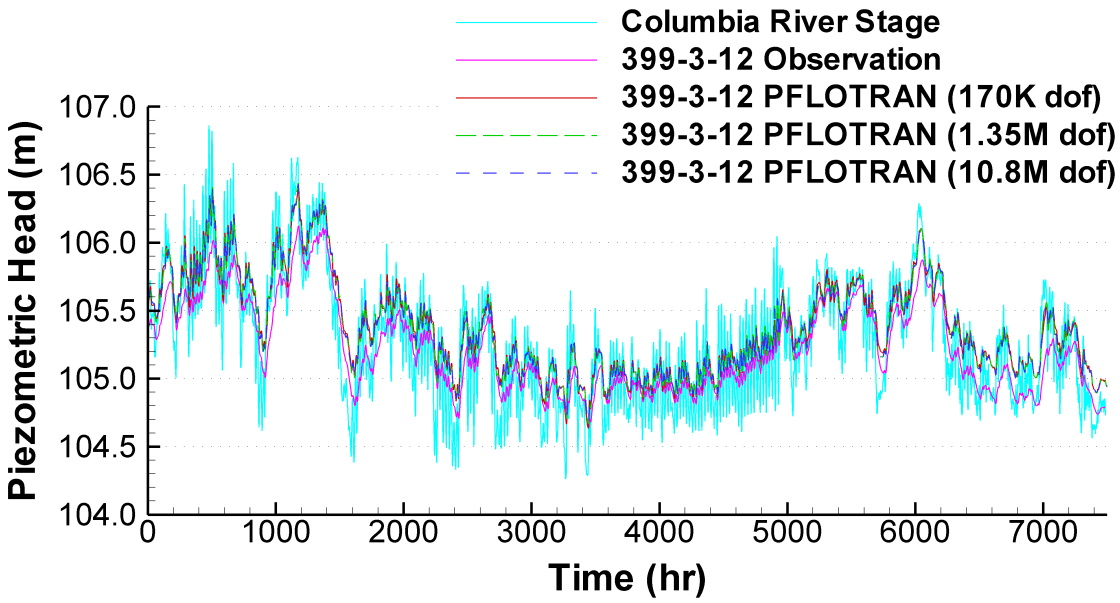

Figure 14: Comparison of observed versus predicted head at Well 399-3-12 with comparison to river stage.

PFLOTRAN grid resolutions produce nearly identical piezometric heads that slightly overestimate the observed head and are more oscillatory.

Figure 16 shows the piezometric head at the center of the Hanford 300 Area IFC site (see Figure 11) as a function of time. All three PFLOTRAN grid resolutions demonstrate nearly identical solutions. Predicted PFLOTRAN pore water flow velocities computed at the center of the IFC site are plotted in Figure 17 for the various grid resolutions. The enlarged plot (Figure 18) reveals that the coarse grid (170K dof) model underestimates peak velocities by as much as 60%, whereas the velocities are more consistent for the 1.35M and 10.8M dof simulations, although it is not clear that velocities have convergence.

Figures 19 and 20 illustrate the magnitude of the components of the pore water velocity at the IFC site over time.

Conclusions

- PFLOTRAN variably-saturated flow simulations utilizing data sets provided by [?, ?, ?] produce piezometric heads slightly higher and more oscillatory than those observed in the field. They also result in a fairly

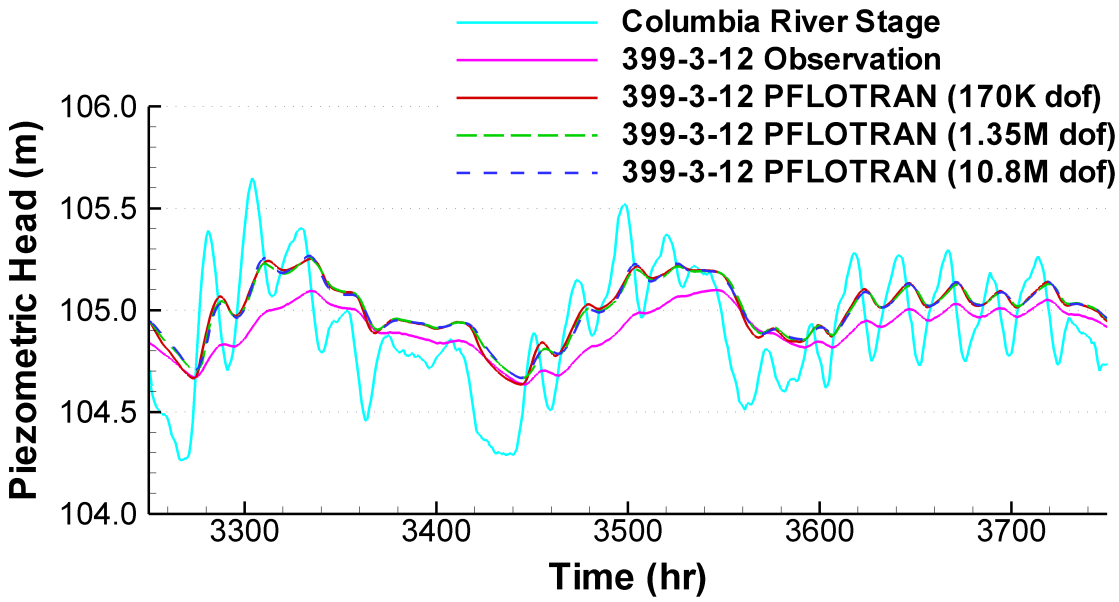

Figure 15: Comparison of observed versus predicted head at Well 399-3-12 with comparison to river stage (enlarged).

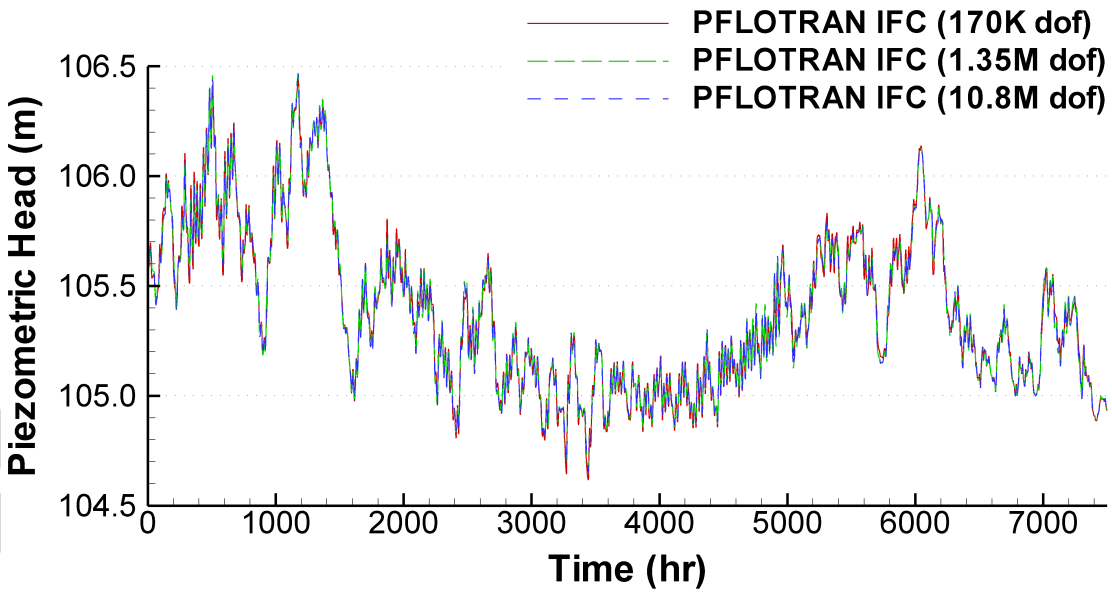

Figure 16: Predicted PFLOTRAN piezometric head at center of Hanford 300 Area IFC site.

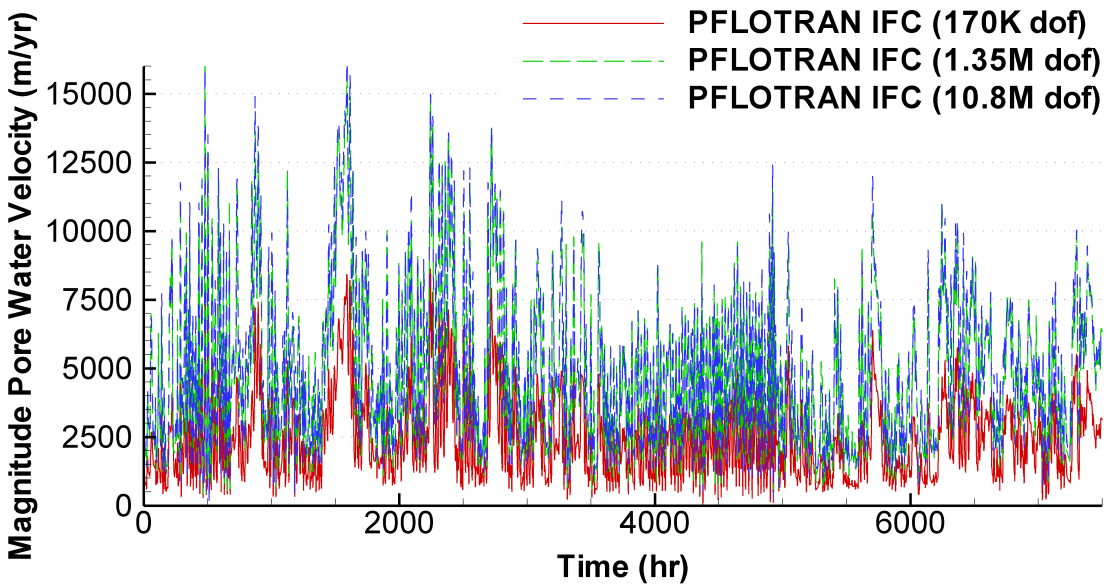

Figure 17: Magnitude of predicted pore water velocities at IFC site (elevation = 100m) based on 20% porosity in Hanford unit.

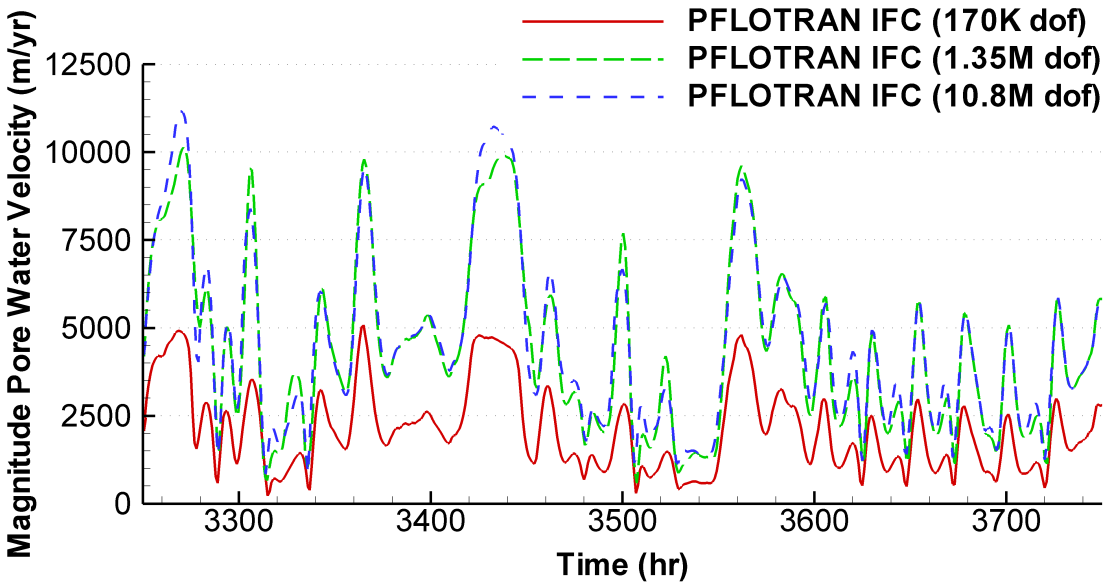

Figure 18: Magnitude of predicted pore water velocities at IFC site (enlarged).

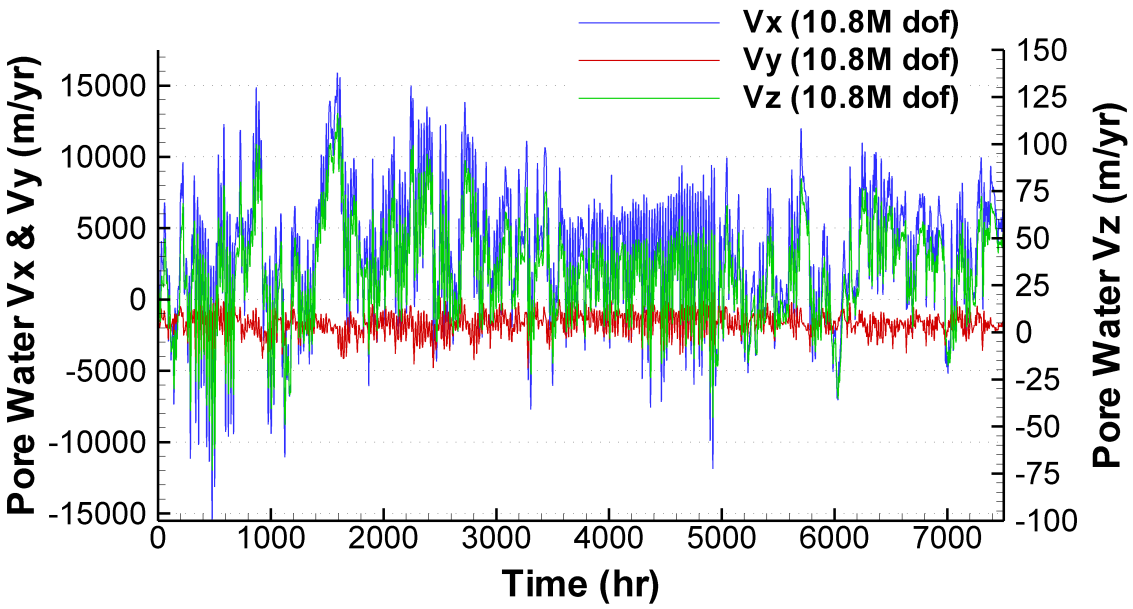

Figure 19: Components of pore water velocity at IFC site (elevation = 100m) based on 20% porosity in Hanford unit.

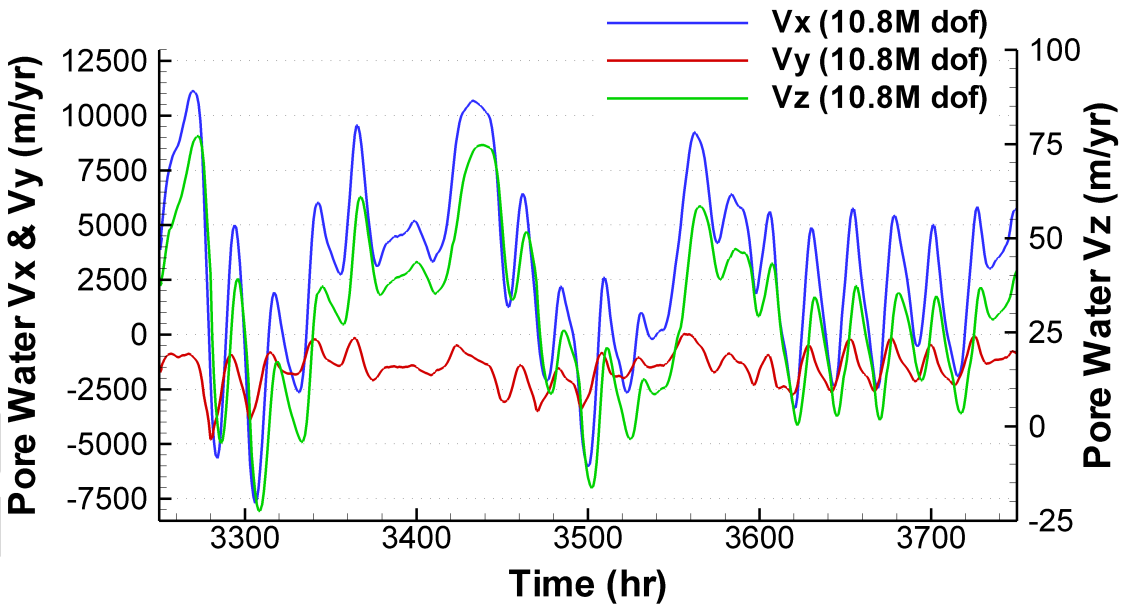

Figure 20: Components of pore water velocity at IFC site (enlarged).

consistent offset from the the observed wells over time (the same holds for other well observation data compared to the predicted heads).

- Calibration of hydraulic parameters (i.e. permeability) and the inland boundary condition is needed to improve agreement between observed and predicted piezometric heads.

- Convergence of flow velocities with higher-resolution grids needs to be investigated further to rule out numerical artifacts in the solution.

- High-performance computation enables the solution of large, 3D high-resolution problem domains beyond what is possible on a single-processor workstation and with reasonable turnaround time, especially for calibration/optimization runs. These high-resolution flow simulations may be necessary to improve the accuracy of flow velocities employed in radionuclide transport simulations.

0.18 Metric Problem

The same conceptual model as described above is used for the metric problem.

0.18.1 Intent

Although computed pressure heads appear to be converged at a relatively coarse grid scale (see figures 17–20), the velocity field is much more sensitive to the grid spacing as would be expected. Therefore, to test convergence and obtain a more accurate flow field, the grid resolution is reduced in the Q2 problem to 2.5 m compared to the finest scale of 5 m previously used. In Q4 the grid was be further refined in the x and y directions by the factor $\sqrt{2}$ giving a factor 2 increase in problem size.

0.18.2 Initial and Boundary Conditions

For these benchmark problems the same initial and boundary conditions will be used as outlined above. However, it will be necessary to compute new hydrostatic initial and boundary conditions on the refined grids.

0.18.3 Application Metric

The metric baseline for this problem is the spatial resolution in the x and y directions. In Q2 we used Δx = Δy = 2.5 m which gives problem size of $540 \times 1000 \times 120 = 64,800,000$. In Q4 we will double the problem size using Δx = Δy = 2.5/$\sqrt{2}$ m = 1.768 m, for a problem size of $764 \times 1414 \times 120$ = 129,635,520, slightly larger than a factor of 2.

0.19 Q2 Metric Status

The metric baseline for this problem is the spatial resolution in the x and y directions. In the Q2 metrics, we used Δx = Δy = 2.5 m which gives problem size of $540 \times 1000 \times 120 = 64,800,000$.

0.19.1 Results

PFLOTRAN was built on the quad-core version of Jaguar partition using the modules `hdf5/1.6.7_par` and`xt-craypat/4.1.1`, in addition to the various modules loadead as part of the default user environment. The executable was built using snapshot `683312f7276d` of the PFLOTRAN development tree, and snapshot `fc1707c94fca` of the PETSc development tree. Optimization flags used were "`-tp barcelona-64 -fastsse`", and BLAS and LAPACK routines were provided by the ACML library.

The $540 \times 1000 \times 120 = 64,800,000$ Q2 baseline benchmark was run on 4000 cores of the Jaguarcnl partition. This number was chosen because it is approximately equal to one-eigth of the anticipated size of the final Jaguar configuration. The benchmark problem was started from a checkpoint file obtained from a previous calculation run to steady-state with a constant average river stage to provide hydrostatic initial conditions. The problem was run for 183 hours of simulation time, which corresponds to 200 flow and 200 transport steps. The maximum permissible time step imposed on the calculation of 1 hour (which is limited by river-stage fluctuations) was reached at step 20 and was used until the end of the run.

Performance data were captured using both the PETSc logging framework and the `pat_hwpc` tool that is part of the CrayPat toolkit. The PETSc logging framework is extremely light-weight, so much so that it was possible to conduct all production runs with it enabled. We also tested and compared `pat_hwpc`-instrumented and non-instrumented runs and found that the overhead introduced by `pat_hwpc` was negligible.

The execution of PFLOTRAN is divided into three PETSc logging stages: "Init", "Time Step", and "Output". The "Init" stage consists of I/O required to read in program options, boundary conditions, geometry and material properties of the problem domain etc., and initialization. The "Time Step" stage consists of the actual computation, and also includes the time spent reading the restart file. The "Output" stage consists of all writes to HDF5, TecPlot, and checkpoint files. The PETSc logging framework reports the following for the various stages:

```
Stage: ----Time---- ----Flops---- ---Messages--- -Message Lengths- -Reductions-
       Avg   %Total    Avg  %Total  counts %Total  Avg     %Total  counts %Total
1     30.632   1.2     0.        0  6.942e5     0  0.248       0     70       0
2    2.5597e3 98.8  1.1386e15 100  1.197e10 100  5.116e3    100   1.092e6  100
3      0.         0  0.          0  0.          0  0.           0      0.      0
------------------------------------------------------------------------------
```

Note that FLOP counts reported are based on "hand-counts", rather than hardware counters, and include only FLOPs that occur inside PETSc routines. In the "Time Step" stage, 17 seconds are spent reading the restart file; the rest of the time spent in that stage is spent in computation. Most of this time is spent in the PETSc `SNESSolve`, which is the driver for the inexact Newton-solver;

its inclusive time is 2531 seconds. Evaluation of nonlinear residuals and Jacobians required 124 seconds (i.e., inside PFLOTRAN "physics" routines), but most of the SNESSolve time (2543 seconds, inclusive) is spent inside the PETSc `KSPSolve`, which is the Krylov subspace method driver (we use BiCGStab in this case). Of this time, `1051` seconds are spent in matrix-vector multiplications, and `824` seconds are spent in applying the preconditioner (block Jacobi with ILU(0) applied on each block).

The `pat_hwpc` tool reports the following performance data (averaged over all processor cores) collected over the entire lifetime of the baseline benchmark run.

```
Experiment=1 / PE='HIDE' / Thread=0='HIDE'

========================================================================
Totals for program
------------------------------------------------------------------------
  Time%                                          100.0%
  Time                                      2594.401863
  Imb.Time                                     0.733177
  Imb.Time%                                       0.0%
  Calls                                            427
  PAPI_TOT_INS             2191.902M/sec  5555623287282 instr
  PAPI_L1_DCA               860.315M/sec  2180567071418 refs
  PAPI_FP_OPS               127.220M/sec   322453892209 ops
  DATA_CACHE_MISSES           4.205M/sec    10658390205 misses
  Cycles                   2534.613 secs  5322688273718 cycles
  User time (approx)       2534.613 secs  5322688273718 cycles
  Utilization rate                               97.7%
  Instr per cycle                                 1.04 inst/cycle
  HW FP Ops / Cycles                              0.06 ops/cycle
  HW FP Ops / User time     127.220M/sec   322453892209 ops   3.0%peak
  HW FP Ops / WCT           124.288M/sec
  HW FP Ops / Inst                                5.8%
  Computation intensity                           0.15 ops/ref
  MIPS                   8767606.37M/sec
  MFLOPS                  508880.58M/sec
  Instructions per LD ST                          2.55 inst/ref
  LD & ST per D1 miss                           204.59 refs/miss
  D1 cache hit ratio                             99.5%
  LD ST per Instructions                         39.2%
========================================================================
```

0.20 Q4 metric status

For the Q4 metric problem, we increased the spatial resolution to $\Delta x = \Delta y = 2.5/\sqrt{2}$ m = 1.768 m, yielding a problem size of $764 \times 1414 \times 120$ = 129,635,520, slightly larger than a factor of 2 times the Q2 problem size. The Q4 benchmark problem was run on 8000 cores of Jaguar (roughly equal to $1/4$ of the total machine). Again, performance data were captured using both the PETSc logging framework and the `pat_hwpc` tool from the CrayPat toolkit.

We describe two different runs (which we will simply denote "Run 1" and "Run 2") of the Q4 benchmark. In Run 1, we employed exactly the same solver algorithms (and the same version of PETSc) as in the Q2 run. In Run 2, the relative tolerance for the inner, linear solve was chosen according to the progress of the outer, nonlinear solver to reduce over-solving; additionally, a newer version of PETSc incorporating an improved version of BiCGStab requiring only

one MPI_Allreduce() was used.

0.20.1 Q4 Run 1

PFLOTRAN was built on the Jaguar partition (quad-core Opteron processors) using the modules hdf5/1.6.7_par and xt-craypat/4.1.1, in addition to various default modules. The executable was built using snapshot 683312f7276d of the PFLOTRAN development tree, and snapshot fc1707c94fca of the PETSc development tree. Optimization flags used were "-tp barcelona-64 -fastsse", and BLAS and LAPACK routines were provided by the ACML library.

The PETSc logging framework reports the following for the previously described execution stages:

```
Stage: ----Time---- ----Flops---- ---Messages--- -Message Lengths- -Reductions-
1     63.029   2.1    0.         0  1.629e6     0  0.2363       0     70       0
2    2.8916e3 98.8  2.5230e15 100  2.527e10 100  5.121e3     100   1.143e6  100
3    2.436e-4  0.0  0.          0  0.          0  0.           0      0.      0
```

Again, FLOP counts reported are based on "hand-counts", rather than hardware counters, and include only those FLOPs that occur inside PETSc routines. In the "Time Step" stage, 34 seconds are spent reading the restart file; the rest of the time spent in that stage is spent in computation. Most of this time is spent in the PETSc SNESSolve, which is the driver for the inexact Newton-solver; its inclusive time is 2844 seconds. Evaluation of nonlinear residuals and Jacobians required 131 seconds (i.e., inside PFLOTRAN "physics" routines), but most of the SNESSolve time (2702 seconds, inclusive) is spent inside the PETSc KSPSolve, which is the Krylov subspace method driver (we use BiCGStab in this case). Of this time, 1072 seconds are spent in matrix-vector multiplications, and 904 seconds are spent in applying the preconditioner (block Jacobi with ILU(0) applied on each block).

The pat_hwpc tool reports the following performance data (averaged over all processor cores) collected over the entire lifetime of the baseline benchmark run.

```
Experiment=1 / PE='HIDE' / Thread=0='HIDE'

========================================================================
Totals for program
------------------------------------------------------------------------
  Time%                                          100.0%
  Time                                      2958.363308
  Imb.Time                                     2.059817
  Imb.Time%                                       0.1%
  Calls                                            427
  PAPI_TOT_INS              2191.132M/sec  6296750964722 instr
  PAPI_L1_DCA                859.657M/sec  2470435150063 refs
  PAPI_FP_OPS                124.417M/sec   357543758456 ops
  DATA_CACHE_MISSES            4.200M/sec    12068627969 misses
  Cycles                    2873.744 secs  6034861553228 cycles
  User time (approx)        2873.744 secs  6034861553228 cycles
  Utilization rate                               97.1%
  Instr per cycle                                 1.04 inst/cycle
  HW FP Ops / Cycles                              0.06 ops/cycle
  HW FP Ops / User time      124.417M/sec   357543758456 ops         3.0%peak
  HW FP Ops / WCT            120.859M/sec
  HW FP Ops / Inst                                5.7%
```

```
  Computation intensity                                 0.14 ops/ref
  MIPS                        17529054.36M/sec
  MFLOPS                        995339.34M/sec
  Instructions per LD ST                                2.55 inst/ref
  LD & ST per D1 miss                                 204.70 refs/miss
  D1 cache hit ratio                                   99.5%
  LD ST per Instructions                               39.2%
========================================================================
```

0.20.2 Q4 Run 2

PFLOTRAN was built on the Jaguar partition (quad-core Opteron processors) using the modules `hdf5/1.6.7_par` and `xt-craypat/4.1.1`, in addition to various default modules. The executable was built using snapshot `c65b3bc13590` of the PFLOTRAN development tree, and snapshot `705586efcbf3` of the PETSc development tree. Optimization flags used were "`-tp barcelona-64 -fastsse`", and BLAS and LAPACK routines were provided by the ACML library.

Q4 Run 2 differs from Q4 Run 1 (and Q2) in two important ways: 1) In Run 2, the "Improved" BiCGStab (IBCGS) method [**?**] is employed for the inner, linear solves. This method is essentially identical to the classical BiCGStab (BCGS) method, but it formulated such that all vector dot products are independent, allowing overlap of communication and computation as well as requiring only one global reduction operation per iteration. This is especially important, because we have found that one of the major factors limiting scalability of PFLOTRAN on Jaguar is the cost of `MPI_Allreduce()` operations in the BiCGStab solve. 2) In Q2 and Q4 Run 1, a constant tolerance for the linear solves inside the Newton solver is employed. This can lead to "oversolving" when the Newton method is far away from the solution (and where the local linear model may disagree considerably from the function whose roots we are trying to find). In Q4 Run 2, we employ a scheme due to Eisenstat and Walker [**?**] ("Choice 1" from the paper, specifically) that attempts to choose a loose tolerance for the linear solves when the Newton method is far from solution, and tightens the tolerance as the Newton method moves closer to the solution. This strategy can dramatically reduce the number of linear solver iterations (where the bulk of the compute time is spent) required during a simulation.

The PETSc logging framework reports the following for the previously described execution stages:

```
Stage: ----Time---- ----Flops---- ---Messages--- -Message Lengths- -Reductions-
1     110.54   8.7    0.         0  7.581e6   0.1  2.822      0.1    104        0
2    1.1536e3 91.3  1.0204e15 100  9.033e9  99.9  5.117e3   99.9  2.136e5   100
3    2.852e-4  0.0  0.           0  0.           0  0.           0      0.      0
```

There is a slight difference in the definition of the stages between this run and Q4 Run 1: The time to read the restart file is now grouped into Stage 1, the initialization stage, rather than being grouped into Stage 2, the Time Step stage. Thus the Time Step stage in Run 2 consists entirely of computation. Most of this time is spent in the PETSc `SNESSolve`, which is the driver for the inexact Newton-solver; its inclusive time is 1140 seconds. Evaluation of nonlinear residuals and Jacobians required 215 seconds (i.e., inside PFLOTRAN "physics" routines), but most of the SNESSolve time is spent inside the PETSc `KSPSolve` (910 seconds, inclusive), which is the Krylov subspace

method driver (we use Improved BiCGStab in this case). Of this time, `330` seconds are spent in matrix-vector multiplications, and `390` seconds are spent in applying the preconditioner (block Jacobi with ILU(0) applied on each block).

The `pat_hwpc` tool reports the following performance data (averaged over all processor cores) collected over the entire lifetime of the baseline benchmark run.

```
Experiment=1 / PE='HIDE' / Thread=0='HIDE'

================================================================================
Totals for program
--------------------------------------------------------------------------------
  Time%                                              100.0%
  Time                                          1267.844830
  Imb.Time                                         1.793267
  Imb.Time%                                           0.1%
  Calls                                                427
  PAPI_L1_DCM                    6.121M/sec     7462665639 misses
  PAPI_TOT_INS                2074.603M/sec  2529330771291 instr
  PAPI_L1_DCA                  860.262M/sec  1048820733496 refs
  PAPI_FP_OPS                  137.731M/sec   167919939236 ops
  User time (approx)          1219.188 secs  2560294807849 cycles
  Cycles                      1219.188 secs  2560294807849 cycles
  User time (approx)          1219.188 secs  2560294807849 cycles
  Utilization rate                                   96.2%
  Instr per cycle                                     0.99 inst/cycle
  HW FP Ops / Cycles                                  0.07 ops/cycle
  HW FP Ops / User time        137.731M/sec   167919939236 ops         1.6%peak
  HW FP Ops / WCT              132.445M/sec
  HW FP Ops / Inst                                    6.6%
  Computation intensity                               0.16 ops/ref
  MIPS                     16596821.91M/sec
  MFLOPS                    1101847.71M/sec
  Instructions per LD ST                              2.41 inst/ref
  LD & ST per D1 miss                               140.54 refs/miss
  D1 cache hit ratio                                 99.3%
  LD ST per Instructions                             41.5%
================================================================================
```

0.21 Metric Interpretation: Q4 to Q2 Comparison

0.21.1 Q4 Run 1 to Q2 weak scaling comparison

Because Q4 Run 1 and the Q2 run employed the exact same algorithms, comparing them essentially amounts to a direct examination of the weak-scaling characteristics of the parallel solvers employed.

Compared to the Q2 run, the Q4 Run 1 used twice the number of processor cores to solve a problem with 2.005 times the total degrees of freedmom, computing 1.133 times more instructions per processor core (2.267 times the total number of instructions) in 1.140 times the wall-clock time. This makes a good case for meeting the Joule weak-scaling metric: The ratio $2.267/1.140 = 1.989$, very close to 2, which would indicate perfect weak scaling. Given that the global reduction operations used extensively in our iterative solvers are not perfectly scalable on the Cray XT4 at this scale, this is excellent.

0.21.2 Q4 Run 2 compared to Q4 Run 1 and Q4 Run 2

Q4 Run 2 took considerably less wall-clock time (1268 seconds) than either Q4 Run 1 (2958 seconds) or the Q2 run (2594 seconds). The Q4 Run 2 employed a signficantly improved version of the BiCGStab solver in PETSc, but the large reduction in wall-clock time is almost entirely due the use of the Eisenstat-Walker strategy for selecting the tolerance for the inner, linear solves, which allows a great reduction in the amount of computation required to conduct the simulation. Q4 Run 2 required only 191674 matrix-vector multiplications compared to 566150 for Q4 Run 1 and 540578 for Q2.

0.21.3 Comparision of Velocity Convergence in Q2 and Q4

One of the goals of being able to run simulations with a greatly increased number of total degrees of freedom is to increase the fidelity of the computed velocity fields. To test the improvement in the solution, the velocity was compared between runs Q2 and Q4. The x- and y-velocities were very similar indicating convergence. However, slight changes were present in the z-velocity, especially in the peak z velocity. This is shown in Figure 21. These peak velocities have the potential of significantly impacting uranium transport at the Hanford 300 Area, the solutions from which are used to assess risk to the environment (i.e. neighboring Columbia River).

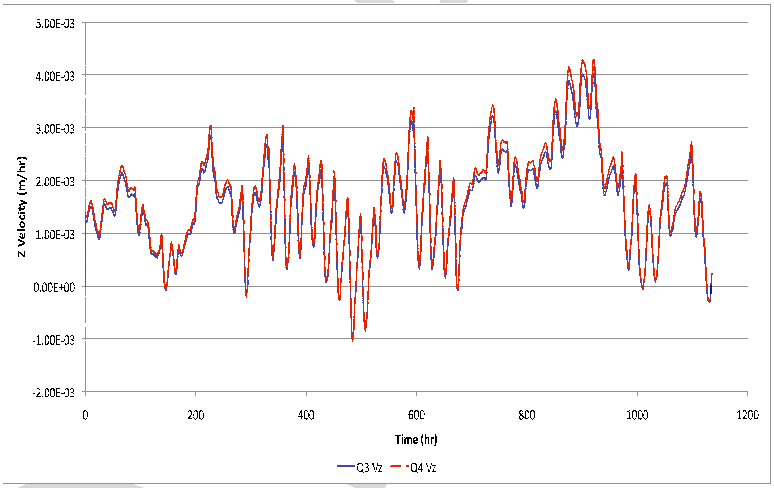

Figure 21: Comparison of the z velocity in Q2 (blue) and Q4 (red) runs.

Appendices

0.22 Appendix: DCA++

0.22.1 Environment of `jaguar.ccs.ornl.gov` for DCA++

```
MODULE_VERSION_STACK=3.1.6
LESSKEY=/etc/lesskey.bin
PAPI_POST_LINK_OPTS= -L/opt/xt-tools/papi/3.5.99cbeta/v22/linux/lib -lpapi -
GNU_VERSION=4.2.1
INFODIR=/usr/local/info:/usr/share/info:/usr/info
MANPATH=/opt/xt-tools/papi/3.5.99cbeta/man:/opt/xt-pe/2.0.44/pe/man:/opt/xt-
2.0.44/mpich2-64/man:/opt/xt-libsci/10.2.0/man:/opt/gcc/4.2.1/cnos/man:/opt/
/2.0.44.ljprokow/usr/man:/opt/xt-os/2.0.44/ros/man:/opt/xt-libc/2.0.44/xt3_g
nome/share/man
HOSTNAME=jaguar13
XKEYSYMDB=/usr/X11R6/lib/X11/XKeysymDB
PAPI_VERSION=3.5.99cbeta
PE_ENV=GNU
_MODULESBEGINENV_=/ccs/home/eisenbac/.modulesbeginenv.jaguar13
HOST=jaguar13
TERM=xterm-color
SHELL=/bin/bash
XTOS_VERSION=2.0.44
PROFILEREAD=true
HISTSIZE=1000
PERFMON_VERSION=v22
SSH_CLIENT=160.91.210.17 63222 22
MPT_DIR=/opt/xt-mpt/2.0.44
BOOT_DIR=/opt/xt-boot/2.0.44
OLDPWD=/tmp/work/eisenbac
PRGENV_DIR=/opt/xt-prgenv/2.0.44
SSH_TTY=/dev/pts/2
ASYNCPE_DIR=/opt/xt-asyncpe/0.1.8
BUILD_OPTS=/opt/xt-pe/2.0.44/bin/snos64/build-opts
USER=eisenbac
LD_LIBRARY_PATH=/opt/xt-tools/papi/3.5.99cbeta/v22/linux/lib:/opt/xt-pe/2.0.
os/lib64:/opt/xt-os/2.0.44/lib:/opt/xt-libc/2.0.44/amd64/lib
LS_COLORS=
TVDSVRLAUNCHCMD=ssh
XNLSPATH=/usr/X11R6/lib/X11/nls
MPICH_DIR=/opt/xt-mpt/2.0.44/mpich2-64/GP
GOTO_NUM_THREADS=1
HOSTTYPE=x86_64
RCLOCAL_PRGENV=true
GCC_VERSION=4.2.1
MPT_VERSION=3.0.0.8
PE_PRODUCT_LIST=MPT:LIBSCI:GCC:GNU:BINUTILS-QUADCORE:XTMPT:ASYNCPE:LUSTRE:PA
PAGER=less
PAPI_INCLUDE_OPTS= -I/opt/xt-tools/papi/3.5.99cbeta/v22/${XTPE_COMPILE_TARGE
OS_DIR=/opt/xt-os/2.0.44
MPICHBASEDIR=/opt/xt-mpt/2.0.44/mpich2-64
CATAMOUNT_DIR=/opt/xt-catamount/2.0.44
```

```
MINICOM=-c on
YOD_LOGFILE=syslog
MODULE_VERSION=3.1.6
MAIL=/var/mail/eisenbac
PATH=/opt/xt-tools/papi/3.5.99cbeta/v22/linux/bin:/opt/xt-pe/2.0.44/cnos/linux/64/bin
ich2-64/GP/bin:/opt/gcc/4.2.1/bin:/opt/moab-5.2.1/bin:/opt/torque/default/bin:/opt/xt
w/usr/sbin:/opt/xt-lustre-ss/2.0.44.ljprokow/usr/bin:/opt/xt-boot/2.0.44/bin/snos64:/
4/bin/snos64:/opt/xt-service/2.0.44/bin/snos64:/opt/xt-prgenv/2.0.44/bin:/sw/xt/hdf5/
usr/X11R6/bin:/bin:/usr/games:/opt/bin:/opt/gnome/bin:/opt/kde3/bin:/opt/pathscale/bi
HDF5_CLIB=-I/sw/xt/hdf5/1.6.5/cnl2.0_pgi7.0.7/include -L/sw/xt/hdf5/1.6.5/cnl2.0_pgi7
lib -lsz -lz
CPU=x86_64
XTPE_COMPILE_TARGET=linux
RCLOCAL_MYSQL=true
INPUTRC=/etc/inputrc
PWD=/ccs/home/eisenbac/Jaguar_3April08
_LMFILES_=/apps/modulefiles/jaguarcnl/hdf5/1.6.5_ser:/opt/modulefiles/PrgEnv-gnu/2.0.
t-mpt-gnu/2.0.44:/opt/modulefiles/xt-libsci/10.2.0:/opt/modulefiles/gcc/4.2.1:/opt/mo
e/2.0.44:/opt/modulefiles/xt-libc/2.0.44:/opt/modulefiles/xt-os/2.0.44:/opt/modulefil
.44:/opt/modulefiles/xt-lustre-ss/2.0.44.ljprokow:/opt/modulefiles/xtpe-target-cnl:/o
-asyncpe/0.1.8:/opt/modulefiles/xt-binutils-quadcore/2.0.0:/opt/modulefiles/torque/2.
opt/modulefiles/xt-papi/3.5.99cbeta
C_DIR=/opt/xt-libc/2.0.44
EDITOR=vi
SYSTEM_USERDIR=/tmp/work/eisenbac
PGI_PRE_COMPILE_OPTS= -Ya,/opt/amd/binutils-070220-amd-sles9/bin
MODULEPATH=/opt/modulefiles:/opt/modules/3.1.6:/apps/modulefiles/jaguarcnl
PATHSCALE_PRE_COMPILE_OPTS= -Ypa,/opt/amd/gnutools-4.2.0-barcelona/bin -gnu4
LIBSCI_INCLUDE_OPTS= -I/opt/acml/default/pgi64/include
LOADEDMODULES=hdf5/1.6.5_ser:PrgEnv-gnu/2.0.44:xt-pe/2.0.44:xt-mpt-gnu/2.0.44:xt-libs
4:xt-libc/2.0.44:xt-os/2.0.44:xt-catamount/2.0.44:xt-boot/2.0.44:xt-lustre-ss/2.0.44
e/0.1.8:xt-binutils-quadcore/2.0.0:torque/2.2.0-snap.200707311754:moab/5.2.1:xt-papi/
MPICH_SMP_SINGLE_COPY_OFF=1
TEXINPUTS=:/ccs/home/eisenbac/.TeX:/usr/share/doc/.TeX:/usr/doc/.TeX
SHLVL=1
HOME=/ccs/home/eisenbac
PTL_SNOS_NAL=SS
LESS_ADVANCED_PREPROCESSOR=no
OSTYPE=linux
SE_DIR=/opt/xt-service/2.0.44
LIBLUSTRE_DEBUG_CONSOLE=0
LS_OPTIONS=-N --color=none -T 0
XCURSOR_THEME=
GCC_PATH=/opt/gcc/4.2.1
GTK_PATH=/usr/local/lib64/gtk-2.0:/opt/gnome/lib64/gtk-2.0:/usr/lib64/gtk-2.0
LESS=-M -I
MACHTYPE=x86_64-suse-linux
LOGNAME=eisenbac
CVS_RSH=ssh
SSH_CONNECTION=160.91.210.17 63222 160.91.205.224 22
MODULESHOME=/opt/modules/3.1.6
```

```
LESSOPEN=lessopen.sh %s
LIBSCI_BASE_DIR=/opt/xt-libsci/10.2.0
HDF5_FLIB=-module . -module /sw/xt/hdf5/1.6.5/cnl2.0_pgi7.0.7/lib -I/sw/xt/h
_pgi7.0.7/lib -lhdf5_fortran -lhdf5 -L/sw/xt/szip/2.0/sles9.2_pgi7.0.7/lib -
INFOPATH=/opt/gcc/4.2.1/cnos/info:/usr/local/info:/usr/share/info:/usr/info
GNU_PRE_COMPILE_OPTS= -B/opt/amd/binutils-070220-amd-sles9/bin
LIBSCI_POST_LINK_OPTS= -L/opt/acml/default/pgi64/lib
PE_DIR=/opt/xt-pe/2.0.44
LESSCLOSE=lessclose.sh %s %s
SCRATCH=/tmp/work/eisenbac
G_BROKEN_FILENAMES=1
LUSTRE_DIR=/opt/xt-lustre-ss/2.0.44.ljprokow
COLORTERM=1
_=/usr/bin/env

Currently Loaded Modulefiles:
  1) hdf5/1.6.5_ser                    11) xt-catamount/2.0.44
  2) PrgEnv-gnu/2.0.44                 12) xt-boot/2.0.44
  3) xt-pe/2.0.44                      13) xt-lustre-ss/2.0.44.ljprokow
  4) xt-mpt-gnu/2.0.44                 14) xtpe-target-cnl
  5) xt-libsci/10.2.0                  15) Base-opts/2.0.44
  6) gcc/4.2.1                         16) xt-asyncpe/0.1.8
  7) xt-mpt/3.0.0.8                    17) xt-binutils-quadcore/2.0.0
  8) xt-service/2.0.44                 18) torque/2.2.0-snap.200707311754
  9) xt-libc/2.0.44                    19) moab/5.2.1
 10) xt-os/2.0.44                      20) xt-papi/3.5.99cbeta
```

0.22.2 Compilation of DCA++ on `jaguar.ccs.ornl.gov`

```
eisenbac@jaguar14:~/dca_14_feb_2008/src> make
perl createversion.pl
CC -O2 -msse3   -I/ccs/home/eisenbac/psimag/include  -DNDEBUG  -DUSE_MPI  -c
/opt/xt-asyncpe/1.0c/bin/CC: INFO: linux target is being used
CC -O2 -msse3   -I/ccs/home/eisenbac/psimag/include  -DNDEBUG  -DUSE_MPI  -c
/opt/xt-asyncpe/1.0c/bin/CC: INFO: linux target is being used
CC -O2 -msse3   -I/ccs/home/eisenbac/psimag/include  -DNDEBUG  -DUSE_MPI  -c
/opt/xt-asyncpe/1.0c/bin/CC: INFO: linux target is being used
CC -O2 -msse3   -I/ccs/home/eisenbac/psimag/include  -DNDEBUG  -DUSE_MPI  -c
/opt/xt-asyncpe/1.0c/bin/CC: INFO: linux target is being used
CC -O2 -msse3   -I/ccs/home/eisenbac/psimag/include  -DNDEBUG  -DUSE_MPI  -c
/opt/xt-asyncpe/1.0c/bin/CC: INFO: linux target is being used
CC -O2 -msse3   -I/ccs/home/eisenbac/psimag/include  -DNDEBUG  -DUSE_MPI  -c
/opt/xt-asyncpe/1.0c/bin/CC: INFO: linux target is being used
CC -O2 -msse3   -I/ccs/home/eisenbac/psimag/include  -DNDEBUG  -DUSE_MPI  -c
/opt/xt-asyncpe/1.0c/bin/CC: INFO: linux target is being used
CC -O2 -msse3   -I/ccs/home/eisenbac/psimag/include  -DNDEBUG  -DUSE_MPI  -c
/opt/xt-asyncpe/1.0c/bin/CC: INFO: linux target is being used
CC -O2 -msse3   -I/ccs/home/eisenbac/psimag/include  -DNDEBUG  -DUSE_MPI  -c
/opt/xt-asyncpe/1.0c/bin/CC: INFO: linux target is being used
CC -O2 -msse3   -I/ccs/home/eisenbac/psimag/include  -DNDEBUG  -DUSE_MPI  -c
```

```
/opt/xt-asyncpe/1.0c/bin/CC: INFO: linux target is being used
CC -O2 -msse3  -c get_tsc.s
/opt/xt-asyncpe/1.0c/bin/CC: INFO: linux target is being used
CC -O2 -msse3  -o dcaqmc_mpi basic.o  init.o main.o  io.o akima.o wofz.o  minimizatio
/opt/xt-asyncpe/1.0c/bin/CC: INFO: linux target is being used
```

0.22.3 Running DCA++ on `jaguar.ccs.ornl.gov`

To execute the binary, one submits the job with the following run script.

```
#! /usr/bin/env bash

##PBS -e __resultsdir__
##PBS -o __resultsdir__
#PBS -N DCA_prof
#PBS -l walltime=1:30:00
#PBS -l size=7824
##PBS -q debug
#PBS -A CSC044

#-----------------------------------------
EXECUTABLE="/tmp/work/eisenbac/dca_joule_q2_double_wo_sigma/dcaqmc_mpi"
WORKDIR="/tmp/work/eisenbac/dca_joule_q2_double_wo_sigma/"
#-----------------------------------------

export PAT_RT_HWPC=1

#--Change directory to the working directory.
cd $WORKDIR

#--Run the executable.
date
time aprun -n 7824 $EXECUTABLE inputDcaPpNc16A_disorder16.inp
date
```

0.23 Appendix: GYRO

0.23.1 Environment of `jaguar.ccs.ornl.gov` for GYRO

```
_=/usr/bin/env
SSH_CONNECTION=128.219.187.8 64981 160.91.205.226 22
LESSCLOSE=lessclose.sh %s %s
PATH=/opt/xt-tools/craypat/4.2beta/v22/bin:/opt/fftw/2.1.5/cnos/bin:
/opt/moab-5.2.1/bin:/opt/torque/default/bin:/opt/xt-asyncpe/0.1.8/bin:
/opt/xt-pe/2.0.44/bin/snos64:/opt/xt-lustre-ss/2.0.44.ljprokow/usr/sbin:
/opt/xt-lustre-ss/2.0.44.ljprokow/usr/bin:/opt/xt-boot/2.0.44/bin/snos64:
/opt/xt-catamount/2.0.44/bin/snos64:/opt/xt-os/2.0.44/bin/snos64:
/opt/xt-service/2.0.44/bin/snos64:/opt/xt-prgenv/2.0.44/bin:
/opt/xt-pe/2.0.44/cnos/linux/64/bin:/opt/pgi/7.1.4/linux86-64/7.1/bin:
/ccs/home/fm9/bin:/usr/local/bin:/usr/bin:/usr/X11R6/bin:/bin:/usr/games:
/opt/bin:/opt/gnome/bin:/opt/kde3/bin:/opt/pathscale/bin:/opt/bin:/opt/publi
/opt/bin:/opt/public/bin:/spin/sys/adm/bin:/ccs/home/fm9/bin:
/ccs/home/fm9/Perl:/spin/sys/adm/bin:/tmp/work/fm9/gyro-7.0/bin
FFTW_DIR=/opt/fftw/2.1.5/cnos/lib
HOSTNAME=jaguar15
XAUTHLOCALHOSTNAME=jaguar15
USER=fm9
LESS_ADVANCED_PREPROCESSOR=no
GTK_PATH=/usr/local/lib64/gtk-2.0:/opt/gnome/lib64/gtk-2.0:/usr/lib64/gtk-2
PGI_PATH=/opt/pgi/7.1.4
SSH_CLIENT=128.219.187.8 64981 22
HOSTTYPE=x86_64
TVDSVRLAUNCHCMD=ssh
ASYNCPE_DIR=/opt/xt-asyncpe/0.1.8
PE_ENV=PGI
FFTW_INC=/opt/fftw/2.1.5/cnos/include
TERM=xterm
INPUTRC=/etc/inputrc
LUSTRE_DIR=/opt/xt-lustre-ss/2.0.44.ljprokow
OS_DIR=/opt/xt-os/2.0.44
G_BROKEN_FILENAMES=1
XCURSOR_THEME=
YOD_LOGFILE=syslog
CPU=x86_64
RCLOCAL_PRGENV=true
INFODIR=/usr/local/info:/usr/share/info:/usr/info
PRGENV_DIR=/opt/xt-prgenv/2.0.44
BOOT_DIR=/opt/xt-boot/2.0.44
SSH_TTY=/dev/pts/1
ENV=/ccs/home/fm9/.kshrc
LS_COLORS=*.f=0;31:*.c=0;35:*.htm=0;33:*.html=0;33
PROFILEREAD=true
MPICH_SMP_SINGLE_COPY_OFF=1
MANPATH=/opt/xt-tools/craypat/4.2beta/v22/../man:/opt/fftw/2.1.5/cnos/man:
/opt/torque/default/man:/opt/mpt/3.0.0.8/xt/man:
/opt/xt-lustre-ss/2.0.44.ljprokow/usr/man:/opt/xt-os/2.0.44/ros/man:
```

```
/opt/xt-libc/2.0.44/xt3_glibc/man:/opt/xt-pe/2.0.44/pe/man:
/opt/xt-pe/2.0.44/papi/man:/opt/xt-libsci/10.2.0/man:
/opt/pgi/7.1.4/linux86-64/7.1/man:/usr/local/man:/usr/share/man:
/usr/X11R6/man:/opt/gnome/share/man
PGI_PRE_COMPILE_OPTS= -Ya,/opt/amd/binutils-070220-amd-sles9/bin
PGI_VERS_STR=7.1.4
GNU_PRE_COMPILE_OPTS= -B/opt/amd/binutils-070220-amd-sles9/bin
PATHSCALE_PRE_COMPILE_OPTS=-Ypa,/opt/amd/gnutools-4.2.0-barcelona/bin -gnu4
CRAYPAT_POST_COMPILE_OPTS= ‘$CRAYPAT_ROOT/sbin/pat-opts POST_COMPILE‘
MINICOM=-c on
SE_DIR=/opt/xt-service/2.0.44
XKEYSYMDB=/usr/X11R6/lib/X11/XKeysymDB
LS_OPTIONS=-N --color=tty -T 0
MAIL=/var/mail/fm9
MODULEPATH=/opt/modulefiles:/opt/modules/3.1.6:/apps/modulefiles/jaguarcnl
LD_LIBRARY_PATH=/opt/fftw/2.1.5/cnos/lib:/opt/xt-os/2.0.44/lib:
/opt/xt-libc/2.0.44/amd64/lib:/opt/xt-pe/2.0.44/lib:
/opt/pgi/7.1.4/linux86-64/7.1/libso:/opt/pgi/7.1.4/linux86-64/7.1/lib
CRAYPAT_PRE_COMPILE_OPTS= ‘$CRAYPAT_ROOT/sbin/pat-opts PRE_COMPILE‘
CRAYPAT_INCLUDE_OPTS= ‘$CRAYPAT_ROOT/sbin/pat-opts INCLUDE‘
DISPLAY=localhost:10.0
LOADEDMODULES=pgi/7.1.4:xt-libsci/10.2.0:xt-mpt/3.0.0.8:xt-pe/2.0.44:
PrgEnv-pgi/2.0.44:xt-service/2.0.44:xt-libc/2.0.44:xt-os/2.0.44:
xt-catamount/2.0.44:xt-boot/2.0.44:xt-lustre-ss/2.0.44.ljprokow:
xtpe-target-cnl:Base-opts/2.0.44:xt-asyncpe/0.1.8:
xt-binutils-quadcore/2.0.0:torque/2.2.0-snap.200707311754:
moab/5.2.1:fftw/2.1.5:xt-craypat/4.2beta
RCLOCAL_MYSQL=true
TEXINPUTS=:/ccs/home/fm9/.TeX:/usr/share/doc/.TeX:/usr/doc/.TeX
MPT_DIR=/opt/mpt/3.0.0.8/xt
MACHTYPE=x86_64-suse-linux
FFTW_INCLUDE_OPTS= -I/opt/fftw/2.1.5/cnos/include
LIBSCI_INCLUDE_OPTS= -I/opt/acml/default/pgi64/include
MODULESHOME=/opt/modules/3.1.6
_MODULESBEGINENV_=/ccs/home/fm9/.modulesbeginenv.jaguar15
PE_DIR=/opt/xt-pe/2.0.44
HOST=jaguar15
INFOPATH=/usr/local/info:/usr/share/info:/usr/info
WORKDIR=/tmp/work/fm9
MPICH_DIR=/opt/mpt/3.0.0.8/xt/mpich2-pgi
MPT_VERSION=3.0.0.8
PERFMON_VERSION=v22
CRAYPAT_VERSION=4.2beta
PGI_VERSION=7.1
XTOS_VERSION=2.0.44
PE_PRODUCT_LIST=BINUTILS-QUADCORE:XTMPT:ASYNCPE:LUSTRE:LIBSCI:PGI:FFTW:CRAYPAT
SYSTEM_USERDIR=/tmp/work/fm9
SHELL=/bin/ksh
MODULE_VERSION=3.1.6
MODULE_VERSION_STACK=3.1.6
GYRO_DIR=/tmp/work/fm9/gyro-7.0
```

```
LIBSCI_BASE_DIR=/opt/xt-libsci/10.2.0
OSTYPE=linux
LESS=-M -I
GOTO_NUM_THREADS=1
CRAYPAT_PRE_LINK_OPTS= '$CRAYPAT_ROOT/sbin/pat-opts PRE_LINK'
CRAYPAT_POST_LINK_OPTS= '$CRAYPAT_ROOT/sbin/pat-opts POST_LINK'
CATAMOUNT_DIR=/opt/xt-catamount/2.0.44
FFTW_POST_LINK_OPTS= -L/opt/fftw/2.1.5/cnos/lib
LIBSCI_POST_LINK_OPTS= -L/opt/acml/default/pgi64/lib
XTPE_COMPILE_TARGET=linux
LESSKEY=/etc/lesskey.bin
LIBLUSTRE_DEBUG_CONSOLE=0
COLORTERM=1
PE_PGI_VARIANT=P2
IDL_PATH=:/tmp/work/fm9/gyro-7.0/vugyro
LM_LICENSE_FILE=/opt/pgi/7.1.4/license.dat
LOGNAME=fm9
BUILD_OPTS=/opt/xt-asyncpe/0.1.8/admin/bin/build-opts
CRAYPAT_ROOT=/opt/xt-tools/craypat/4.2beta/v22/cpatx
$USER@jaguar15>
PGI=/opt/pgi/7.1.4
PYTHONPATH=/tmp/work/fm9/gyro-7.0/python
CVSROOT=:local:/spin/sys/adm/cvs
C_DIR=/opt/xt-libc/2.0.44
GYRO_PLAT=XT_CNL
HISTSIZE=2500
HOME=/ccs/home/fm9
_LMFILES_=/opt/modulefiles/pgi/7.1.4:/opt/modulefiles/xt-libsci/10.2.0:
/opt/modulefiles/xt-mpt/3.0.0.8:/opt/modulefiles/xt-pe/2.0.44:
/opt/modulefiles/PrgEnv-pgi/2.0.44:/opt/modulefiles/xt-service/2.0.44:
/opt/modulefiles/xt-libc/2.0.44:/opt/modulefiles/xt-os/2.0.44:
/opt/modulefiles/xt-catamount/2.0.44:/opt/modulefiles/xt-boot/2.0.44:
/opt/modulefiles/xt-lustre-ss/2.0.44.ljprokow:/opt/modulefiles/xtpe-target-c
/opt/modulefiles/Base-opts/2.0.44:/opt/modulefiles/xt-asyncpe/0.1.8:
/opt/modulefiles/xt-binutils-quadcore/2.0.0:
/opt/modulefiles/torque/2.2.0-snap.200707311754:/opt/modulefiles/moab/5.2.1
/opt/modulefiles/fftw/2.1.5:/opt/modulefiles/xt-craypat/4.2beta
LESSOPEN=lessopen.sh %s
OS=Linu
PTL_SNOS_NAL=SS
XNLSPATH=/usr/X11R6/lib/X11/nls
CVS_RSH=ssh
PAGER=less
Currently Loaded Modulefiles:
  1) pgi/7.1.4                        11) xt-lustre-ss/2.0.44.ljprokow
  2) xt-libsci/10.2.0                 12) xtpe-target-cnl
  3) xt-mpt/3.0.0.8                   13) Base-opts/2.0.44
  4) xt-pe/2.0.44                     14) xt-asyncpe/0.1.8
  5) PrgEnv-pgi/2.0.44                15) xt-binutils-quadcore/2.0.0
  6) xt-service/2.0.44                16) torque/2.2.0-snap.200707311754
  7) xt-libc/2.0.44                   17) moab/5.2.1
```

```
  8) xt-os/2.0.44                  18) fftw/2.1.5
  9) xt-catamount/2.0.44           19) xt-craypat/4.2beta
 10) xt-boot/2.0.44
```

0.23.2 Compilation of GYRO on `jaguar.ccs.ornl.gov`

```
cd /tmp/work/fm9/gyro-7.0/ORB ; \
        gmake "FFLAGS = -O3 -Msmartalloc"; \
cd /tmp/work/fm9/gyro-7.0/BLEND ; \
        gmake "FFLAGS = -O3 -Msmartalloc"; \
cd /tmp/work/fm9/gyro-7.0/math ; \
        gmake "FFLAGS = -O3 -Msmartalloc"; \
cd /tmp/work/fm9/gyro-7.0/TRANSPOSE ; \
        gmake "FFLAGS = -O3 -Msmartalloc"; \
cd /tmp/work/fm9/gyro-7.0/SSUB ; \
        gmake "FFLAGS = -O3 -Msmartalloc"; \
cd /tmp/work/fm9/gyro-7.0/GEO ; \
        gmake "FFLAGS = -O3 -Msmartalloc"; \
cd /tmp/work/fm9/gyro-7.0/UMFPACK ; \
        gmake "FFLAGS = -O3 -Msmartalloc"; \
cd /tmp/work/fm9/gyro-7.0/src ; \
gmake "FFLAGS = -O3 -Msmartalloc"
gmake[1]: Entering directory '/lustre/scr72a/fm9/gyro-7.0/ORB'
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
ORB_private.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
ORB_init.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
ORB_lambda.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
ORB_do.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
ORB_s2lambda.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
ORB_s.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ar cr ORB_lib.a ORB_private.o ORB_init.o ORB_lambda.o ORB_do.o
ORB_s2lambda.o ORB_s.o
gmake[1]: Leaving directory '/lustre/scr72a/fm9/gyro-7.0/ORB'
gmake[1]: Entering directory '/lustre/scr72a/fm9/gyro-7.0/BLEND'
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
BLEND_private.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
BLEND_init.f90
```

```
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
BLEND_do.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
BLEND_F.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
BLEND_Fp.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
BLEND_f2.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
BLEND_f2p.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
BLEND_f3.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
BLEND_f3p.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
BLEND_f4.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
BLEND_f4p.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
BLEND_cleanup.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ar cr BLEND_lib.a BLEND_private.o BLEND_init.o BLEND_do.o BLEND_F.o
BLEND_Fp.o BLEND_f2.o BLEND_f2p.o BLEND_f3.o BLEND_f3p.o BLEND_f4.o
BLEND_f4p.o BLEND_cleanup.o
gmake[1]: Leaving directory ‘/lustre/scr72a/fm9/gyro-7.0/BLEND’
gmake[1]: Entering directory ‘/lustre/scr72a/fm9/gyro-7.0/math’
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
math_constants.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn  -O3 -Msmartalloc -c j0y0.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn  -O3 -Msmartalloc -c i0.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn  -O3 -Msmartalloc -c erf.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn  -O3 -Msmartalloc -c proc_time.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
p32.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
```

```
invert_p32.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
gauss_legendre.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
polydiff.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
poly2diff.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
pascal.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
bound_deriv.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
remap_grid.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn  -O3 -Msmartalloc -c rjbesl.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn  -O3 -Msmartalloc -c ribesl.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
cub_spline.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn  -O3 -Msmartalloc -c gamma.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
energy_integral.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
neo_theory.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ar cr math_lib.a math_constants.o j0y0.o i0.o erf.o proc_time.o p32.o
invert_p32.o gauss_legendre.o polydiff.o poly2diff.o pascal.o
bound_deriv.o remap_grid.o rjbesl.o ribesl.o cub_spline.o gamma.o
energy_integral.o neo_theory.o
gmake[1]: Leaving directory `/lustre/scr72a/fm9/gyro-7.0/math'
gmake[1]: Entering directory `/lustre/scr72a/fm9/gyro-7.0/TRANSPOSE'
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
fTRANSP_GLOBALS.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
fTRANSP_INIT.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
fTRANSP_DO.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
```

```
fTRANSP_CLEANUP.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
rTRANSP_GLOBALS.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
rTRANSP_INIT.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
rTRANSP_DO.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
rTRANSP_CLEANUP.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ar cr TRANSP_lib.a fTRANSP_GLOBALS.o fTRANSP_INIT.o fTRANSP_DO.o
fTRANSP_CLEANUP.o rTRANSP_GLOBALS.o rTRANSP_INIT.o rTRANSP_DO.o
rTRANSP_CLEANUP.o
gmake[1]: Leaving directory '/lustre/scr72a/fm9/gyro-7.0/TRANSPOSE'
gmake[1]: Entering directory '/lustre/scr72a/fm9/gyro-7.0/SSUB'
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
SSUB_private.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
fSSUB.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
rSSUB.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
SSUB_init.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
SSUB_cleanup.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ar cr SSUB_lib.a SSUB_private.o fSSUB.o rSSUB.o SSUB_init.o
SSUB_cleanup.o
gmake[1]: Leaving directory '/lustre/scr72a/fm9/gyro-7.0/SSUB'
gmake[1]: Entering directory '/lustre/scr72a/fm9/gyro-7.0/GEO'
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
GEO_interface.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
GEO_alloc.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
GEO_do.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
GEO_interp.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
```

```
GEO_write.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ar cr GEO_lib.a GEO_interface.o GEO_alloc.o GEO_do.o GEO_interp.o
GEO_write.o
gmake[1]: Leaving directory '/lustre/scr72a/fm9/gyro-7.0/GEO'
gmake[1]: Entering directory '/lustre/scr72a/fm9/gyro-7.0/UMFPACK'
ftn   -O3 -Msmartalloc -c umd2fb.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2co.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2fa.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2f0.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2f1.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2f2.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2fg.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2in.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2of.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2s2.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2sl.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2so.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2su.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2er.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2p1.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2p2.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2lt.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2ut.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2rf.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2ra.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2r0.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2r2.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd2rg.f
```

```
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umd21i.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2fb.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2co.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2fa.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2f0.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2f1.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2f2.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2fg.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2in.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2of.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2s2.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2sl.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2so.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2su.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2er.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2p1.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2p2.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2rf.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2ra.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2r0.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2r2.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz2rg.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c umz21i.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c mc21b.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn   -O3 -Msmartalloc -c mc13e.f
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
```

```
ar cr UMFPACK_lib.a umd2fb.o umd2co.o umd2fa.o umd2f0.o umd2f1.o
umd2f2.o umd2fg.o umd2in.o umd2of.o umd2s2.o umd2sl.o umd2so.o
umd2su.o umd2er.o umd2p1.o umd2p2.o umd2lt.o umd2ut.o umd2rf.o
umd2ra.o umd2r0.o umd2r2.o umd2rg.o umd21i.o umz2fb.o umz2co.o
umz2fa.o umz2f0.o umz2f1.o umz2f2.o umz2fg.o umz2in.o umz2of.o
umz2s2.o umz2sl.o umz2so.o umz2su.o umz2er.o umz2p1.o umz2p2.o
umz2rf.o umz2ra.o umz2r0.o umz2r2.o umz2rg.o umz21i.o mc21b.o mc13e.o
gmake[1]: Leaving directory '/lustre/scr72a/fm9/gyro-7.0/UMFPACK'
gmake[1]: Entering directory '/lustre/scr72a/fm9/gyro-7.0/src'
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
gyro_globals.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
pointers.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
nl_private.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
collision_private.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
maxwell_private.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
poisson_private.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
gyro_private.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
profile_exp_private.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc
-I/sw/xt/mumps/4.7.3/cnl2.0_pgi7.0.7_par/include -o mumps_private.o -c
mumps_private.mumps.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
umfpack_private.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
sparse_solve_umfpack.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc
-I/sw/xt/mumps/4.7.3/cnl2.0_pgi7.0.7_par/include -o
sparse_solve_mumps.o -c sparse_solve_mumps.mumps.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_vel_sum_p.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
```

```
get_vel_sum_a.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
zero_check.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
select_methods.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
initialize_timestep.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
initialize_arrays.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
set_profile_miller.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
setup_nonlinear_emf.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
filelog.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
b_bounce_pt.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
b_norm.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
gyro_ave.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
gyro_bdoubleave.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
catch_error.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
catch_blowup.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
send_message_real.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
send_message.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
send_line.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
```

```
catch_halt_signal.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
parallel_dim.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
collect_complex.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
collect_real.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
collect_integer.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
alloc_add.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
allocate_big.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
allocate_profile_exp.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
timestep_SSP_322.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
timestep_explicit.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
timestep_explicit_neo5.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
timestep_SSP_322_neo5.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
maxwell_p.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
maxwell_match.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
poisson_p.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
poisson_match.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
ampere_p.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
```

```
ampere_match.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
map_experimental_profiles.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_maxwell_matrix.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_MPI_grid.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_blend_arrays.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_blend_arrays_emf.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_collision.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_collision_stencil.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_theta_grid.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_omegas.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_omegas_neo.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_profiles.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_profile_1.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_profile_2.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_profile_3.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_profile_4.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_profile_5.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
```

```
make_profile_6.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_profile_8.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_profile_9.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_experimental_profiles.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_experimental_neo_profiles.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_geometry_arrays.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_radial_operators.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_theta_operators.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_gyro.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_lambda_grid.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_phase_space.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_pointers.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_pointer_dimensions.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_dharmonic.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_initial_h.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_initial_neo_h.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_radial_fc.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
```

```
make_simulation_box.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_implicit_advect.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_coll_vel_matrix.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_coll_ampere_matrix.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_poisson_matrix.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_ampere_blend.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_poisson_blend.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
make_phi_doppler.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -o
make_nl.o -c make_nl.fftw.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_nonlinear_advance.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_nonlinear_transfer.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_ampere_coll.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_f_coll.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_gyro_h.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_storage.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_maxwell_solution.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_poisson_solution.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
```

```
get_poisson_explicit.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_ampere_explicit.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_RHS.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_RHS_iso_neo.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_moments_plot.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_field_spectrum.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_field_plot.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_field_r0_plot.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_nonlinear_flux_passing.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_nonlinear_flux_trapped.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_neo_flux_current.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_a_dot_dop_emf.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_nonlinear_emf.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_emf.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_field_fluxave.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_diffusivity.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_neo_diffusivity.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
```

```
get_neo_std_diffusivity.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_zonal_moments.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_field_explicit.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_adiabatic_advance.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_kinetic_advance.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_delta_he.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_delta_he_neo.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_he.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_field_interpolation.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_error.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_entropy.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_phi_squared.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_g_squared.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
do_adaptive_source.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
do_collision.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
do_dtau.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
do_dtau_neo.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
```

```
do_fulladvance.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
do_fulladvance_neo5.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
do_gyro.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
do_nlfast_p.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
do_nlfast_match.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -o
do_nlfft_p.o -c do_nlfft_p.fftw.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -o
do_nlfft_match.o -c do_nlfft_match.fftw.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
do_neo_collision.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
do_pitch_angle_scatter.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
do_pitch_angle_deriv.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
do_ion_conserve.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
do_collision_correct.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
do_collision_correct_neo.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
do_collision_neo5.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
do_pnl.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
do_pnl_correct.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
read_input.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
```

```
read_input_extra.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
readbc_int.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
readbc_real.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
read_experimental_profiles.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
bcast_experimental_profiles.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
rescale_experimental_profiles.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -o
read_restart.o -c read_restart.mpiio.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_big.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_step.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_input.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_theta_operators.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_radial_operators.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_freq.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_field.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_field_rms.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_field_r0.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_field_spectrum.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
```

```
write_ballooning_mode.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_nonlinear_flux.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_nonlinear_transfer_n.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_turbulent_energy_n.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_diffusivity.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_diffusivity_trapped.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_sp_diffusivity.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_diffusivity_i.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_diffusivity_i_ch.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_diffusivity_i_trapped.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_sp_diffusivity_i.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_emf_i.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_neo_diffusivity.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_neo_diffusivity_i.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_h.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_zonal_moments.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_zerobar.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
```

```
write_moments.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_collision.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_adaptive_source.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_efficiency.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_profile_sim.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_profile_exp.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_profile_vugyro.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_timing.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_linear_summary.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_emf_profile.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -o
write_restart.o -c write_restart.mpiio.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_error.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_entropy.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_prec.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_diffusivity_n.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_phi_squared_QL_n.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_g_squared_QL_n.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
```

```
write_geometry_arrays.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_pnl.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_velocity.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
write_matrix_stat.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
cleanup_gyro.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc
-I/sw/xt/mumps/4.7.3/cnl2.0_pgi7.0.7_par/include -o cleanup_mumps.o -c
cleanup_mumps.mumps.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ar cr gyro_lib.a gyro_globals.o pointers.o nl_private.o
collision_private.o maxwell_private.o poisson_private.o gyro_private.o
profile_exp_private.o mumps_private.o umfpack_private.o
sparse_solve_umfpack.o sparse_solve_mumps.o get_vel_sum_p.o
get_vel_sum_a.o zero_check.o select_methods.o initialize_timestep.o
initialize_arrays.o set_profile_miller.o setup_nonlinear_emf.o
filelog.o b_bounce_pt.o b_norm.o gyro_ave.o gyro_bdoubleave.o
catch_error.o catch_blowup.o send_message_real.o send_message.o
send_line.o catch_halt_signal.o parallel_dim.o collect_complex.o
collect_real.o collect_integer.o alloc_add.o allocate_big.o
allocate_profile_exp.o timestep_SSP_322.o timestep_explicit.o
timestep_explicit_neo5.o timestep_SSP_322_neo5.o maxwell_p.o
maxwell_match.o poisson_p.o poisson_match.o ampere_p.o ampere_match.o
map_experimental_profiles.o make_maxwell_matrix.o make_MPI_grid.o
make_blend_arrays.o make_blend_arrays_emf.o make_collision.o
make_collision_stencil.o make_theta_grid.o make_omegas.o
make_omegas_neo.o make_profiles.o make_profile_1.o make_profile_2.o
make_profile_3.o make_profile_4.o make_profile_5.o make_profile_6.o
make_profile_8.o make_profile_9.o make_experimental_profiles.o
make_experimental_neo_profiles.o make_geometry_arrays.o
make_radial_operators.o make_theta_operators.o make_gyro.o
make_lambda_grid.o make_phase_space.o make_pointers.o
make_pointer_dimensions.o make_dharmonic.o make_initial_h.o
make_initial_neo_h.o make_radial_fc.o make_simulation_box.o
make_implicit_advect.o make_coll_vel_matrix.o
make_coll_ampere_matrix.o make_poisson_matrix.o make_ampere_blend.o
make_poisson_blend.o make_phi_doppler.o make_nl.o
get_nonlinear_advance.o get_nonlinear_transfer.o get_ampere_coll.o
get_f_coll.o get_gyro_h.o get_storage.o get_maxwell_solution.o
get_poisson_solution.o get_poisson_explicit.o get_ampere_explicit.o
get_RHS.o get_RHS_iso_neo.o get_moments_plot.o get_field_spectrum.o
get_field_plot.o get_field_r0_plot.o get_nonlinear_flux_passing.o
get_nonlinear_flux_trapped.o get_neo_flux_current.o
```

```
get_a_dot_dop_emf.o get_nonlinear_emf.o get_emf.o get_field_fluxave.o
get_diffusivity.o get_neo_diffusivity.o get_neo_std_diffusivity.o
get_zonal_moments.o get_field_explicit.o get_adiabatic_advance.o
get_kinetic_advance.o get_delta_he.o get_delta_he_neo.o get_he.o
get_field_interpolation.o get_error.o get_entropy.o get_phi_squared.o
get_g_squared.o do_adaptive_source.o do_collision.o do_dtau.o
do_dtau_neo.o do_fulladvance.o do_fulladvance_neo5.o do_gyro.o
do_nlfast_p.o do_nlfast_match.o do_nlfft_p.o do_nlfft_match.o
do_neo_collision.o do_pitch_angle_scatter.o do_pitch_angle_deriv.o
do_ion_conserve.o do_collision_correct.o do_collision_correct_neo.o
do_collision_neo5.o do_pnl.o do_pnl_correct.o read_input.o
read_input_extra.o readbc_int.o readbc_real.o
read_experimental_profiles.o bcast_experimental_profiles.o
rescale_experimental_profiles.o read_restart.o write_big.o
write_step.o write_input.o write_theta_operators.o
write_radial_operators.o write_freq.o write_field.o write_field_rms.o
write_field_r0.o write_field_spectrum.o write_ballooning_mode.o
write_nonlinear_flux.o write_nonlinear_transfer_n.o
write_turbulent_energy_n.o write_diffusivity.o
write_diffusivity_trapped.o write_sp_diffusivity.o
write_diffusivity_i.o write_diffusivity_i_ch.o
write_diffusivity_i_trapped.o write_sp_diffusivity_i.o write_emf_i.o
write_neo_diffusivity.o write_neo_diffusivity_i.o write_h.o
write_zonal_moments.o write_zerobar.o write_moments.o
write_collision.o write_adaptive_source.o write_efficiency.o
write_profile_sim.o write_profile_exp.o write_profile_vugyro.o
write_timing.o write_linear_summary.o write_emf_profile.o
write_restart.o write_error.o write_entropy.o write_prec.o
write_diffusivity_n.o write_phi_squared_QL_n.o write_g_squared_QL_n.o
write_geometry_arrays.o write_pnl.o write_velocity.o
write_matrix_stat.o cleanup_gyro.o cleanup_mumps.o
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
BigScience_globals.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
BigScience.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -r8  -O3 -Msmartalloc -c
get_inputpath.f90
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
ftn -module /tmp/work/fm9/gyro-7.0/modules -O3 -Msmartalloc -o
BigScience BigScience_globals.o BigScience.o get_inputpath.o
/tmp/work/fm9/gyro-7.0/src/gyro_lib.a
/tmp/work/fm9/gyro-7.0/ORB/ORB_lib.a
/tmp/work/fm9/gyro-7.0/BLEND/BLEND_lib.a
/tmp/work/fm9/gyro-7.0/math/math_lib.a
/tmp/work/fm9/gyro-7.0/TRANSPOSE/TRANSP_lib.a
/tmp/work/fm9/gyro-7.0/SSUB/SSUB_lib.a
/tmp/work/fm9/gyro-7.0/GEO/GEO_lib.a
/tmp/work/fm9/gyro-7.0/UMFPACK/UMFPACK_lib.a
/tmp/work/fm9/gyro-7.0/src/gyro_lib.a -ldrfftw -ldfftw -L
```

```
/sw/xt/mumps/4.7.3/cnl2.0_pgi7.0.7_par/lib -lzmumps -lpord
/opt/xt-asyncpe/0.1.8/bin/ftn: INFO: linux target is being used
gmake[1]: Leaving directory '/lustre/scr72a/fm9/gyro-7.0/src'

cd src
pat_build -w BigScience
cd ..
```

0.23.3 Running GYRO on `jaguar.ccs.ornl.gov`

To execute the binary, one submits the job with the following run script.

```
#PBS -N c32x32.B.m30
#PBS -A CSC044
#PBS -j oe
#PBS -o batch.out
#PBS -l walltime=1:00:00,size=4608

cd $PBS_O_WORKDIR

. ${MODULESHOME}/init/ksh
module purge
module load PrgEnv-pgi Base-opts
module load xt-asyncpe/0.1.8 xtpe-target-cnl
module swap xt-mpt xt-mpt/3.0.0.8
module load xt-binutils-quadcore
module load torque moab
export MPICH_SMP_SINGLE_COPY_OFF=1

module load xt-craypat/4.2beta

export PAT_RT_HWPC=0

# 20 timesteps
cp input.dat.20 input.dat
aprun -n 4608 ../../src/BigScience+pat
cp prec.out prec.out.20
cp timing.out timing.out.20

# 10 timesteps
cp input.dat.10 input.dat
aprun -n 4608 ../../src/BigScience+pat
cp prec.out prec.out.10
cp timing.out timing.out.10
```

0.24 Appendix: PFLOTRAN

0.24.1 Environment of `jaguar.ccs.ornl.gov` for PFLOTRAN

```
pflotran_c65b3bc13590/src/pflotran> export PETSC_ARCH=cray-xt4-pgi_craypat
pflotran_c65b3bc13590/src/pflotran> export PETSC_DIR=/ccs/proj/geo002/petsc-
pflotran_c65b3bc13590/src/pflotran> module list

Currently Loaded Modulefiles:

  1) modules/3.1.6                    14) xt-libc/2.0.49a
  2) DefApps                          15) xt-os/2.0.49a
  3) torque/2.2.0-snap.200707311754   16) xt-catamount/2.0.49a
  4) moab/5.2.3                       17) xt-boot/2.0.49a
  5) xt-binutils-quadcore/2.0.1       18) xt-lustre-ss/2.0.49a
  6) MySQL/4.0.27                     19) xtpe-target-cnl
  7) pgi/7.1.6                        20) Base-opts/2.0.49a
  8) xt-libsci/10.2.1                 21) xtpe-quadcore
  9) xt-mpt/3.0.0                     22) hdf5/1.6.7_par
 10) xt-pe/2.0.49a                    23) python/2.5.2-alt
 11) xt-asyncpe/1.0c                  24) mercurial/1.0
 12) PrgEnv-pgi/2.0.49a               25) xt-craypat/4.1.1
 13) xt-service/2.0.49a

jaguar10:proj/pflotran_c65b3bc13590/src/pflotran> env
LESSKEY=/etc/lesskey.bin
MODULE_VERSION_STACK=3.1.6
MANPATH=/opt/xt-tools/craypat/4.1.1/man:
/sw/xt/mercurial/1.0/sles9.2_gnu4.2.1/man:/opt/xt-lustre-ss/2.0.49a/usr/man
/opt/xt-os/2.0.49a/ros/man:/opt/xt-libc/2.0.49a/xt3_glibc/man:
/opt/xt-pe/2.0.49a/papi/man:/opt/mpt/3.0.0/xt/man:/opt/xt-libsci/10.2.1/man
/opt/pgi/7.1.6/linux86-64/7.1/man:/opt/MySQL/4.0.27/man:
/opt/torque/default/man:/usr/local/man:/usr/share/man:/usr/X11R6/man:
/opt/gnome/share/man:/opt/xt-pe/2.0.49a/pe/man
INFODIR=/usr/local/info:/usr/share/info:/usr/info
HOSTNAME=jaguar10
XKEYSYMDB=/usr/X11R6/lib/X11/XKeysymDB
_MODULESBEGINENV_=/ccs/home/rmills/.modulesbeginenv.jaguar10
PATHSCALE_POST_COMPILE_OPTS= -Ypa,/opt/amd/gnutools-4.2.0-barcelona/bin
-gnu4 -Ypa,/opt/amd/gnutools-4.2.0-barcelona/bin -gnu4 -march=barcelona
-Ypa,/opt/amd/gnutools-4.2.0-barcelona/bin -gnu4
PE_ENV=PGI
SHELL=/bin/bash
TERM=xterm
HOST=jaguar10
ASSEMBLER_X86_64=/opt/amd/binutils-070220-amd-sles9/bin/as
HISTSIZE=1000
PROFILEREAD=true
XTOS_VERSION=2.0.49a
SSH_CLIENT=128.219.176.51 50923 22
PETSC_ARCH=cray-xt4-pgi_craypat
```

```
MPT_DIR=/opt/mpt/3.0.0/xt
CRAY_POST_COMPILE_OPTS= -hcpu=barcelona
LIBRARY_PATH=/sw/xt/python/2.5.2/sles9.2_gnu3.3.3//lib
CVSROOT=/ccs/home/rmills/cvsroot
FPATH=/sw/xt/netcdf/3.6.2/sles9.2_gnu4.2.1-alt/include:
/sw/xt/python/2.5.2/sles9.2_gnu3.3.3//include
BOOT_DIR=/opt/xt-boot/2.0.49a
SSH_TTY=/dev/pts/5
PRGENV_DIR=/opt/xt-prgenv/2.0.49a
CRAYPAT_POST_COMPILE_OPTS= ‘$PAT_ROOT/sbin/pat-opts POST_COMPILE‘
USER=rmills
BUILD_OPTS=/opt/xt-asyncpe/1.0c/bin/build-opts
ASYNCPE_DIR=/opt/xt-asyncpe/1.0c
SVN_EDITOR=vim
LD_LIBRARY_PATH=/sw/xt/netcdf/3.6.2/sles9.2_gnu4.2.1-alt/lib:
/sw/xt/python/2.5.2/sles9.2_gnu3.3.3//lib/:/opt/xt-os/2.0.49a/lib:
/opt/xt-libc/2.0.49a/amd64/lib:/opt/xt-pe/2.0.49a/lib:
/opt/pgi/7.1.6/linux86-64/7.1/libso:/opt/pgi/7.1.6/linux86-64/7.1/lib:
/opt/MySQL/4.0.27/lib/mysql
LC_PE_ENV=pgi
XNLSPATH=/usr/X11R6/lib/X11/nls
TVDSVRLAUNCHCMD=ssh
MPICH_DIR=/opt/mpt/3.0.0/xt/mpich2-pgi
PGI_VERS_STR=7.1.6
HOSTTYPE=x86_64
GOTO_NUM_THREADS=1
CPATH=/sw/xt/netcdf/3.6.2/sles9.2_gnu4.2.1-alt/include:
/sw/xt/python/2.5.2/sles9.2_gnu3.3.3//include
RCLOCAL_PRGENV=true
PE_PRODUCT_LIST=XTPE_QUADCORE:LUSTRE:ASYNCPE:XTMPT:LIBSCI:PGI:BINUTILS-QUADCORE:CRAYP
MPT_VERSION=3.0.0
PAGER=less
PGI_VERSION=7.1
OS_DIR=/opt/xt-os/2.0.49a
MPICHBASEDIR=/opt/mpt/3.0.0/xt
MINICOM=-c on
CATAMOUNT_DIR=/opt/xt-catamount/2.0.49a
PGI=/opt/pgi/7.1.6
PATH=/proj/perc/TOOLS/tau_latest/x86_64/bin:./:/ccs/home/rmills/bin:
/spin/sys/adm/bin:/opt/xt-tools/craypat/4.1.1/bin:
/sw/xt/mercurial/1.0/sles9.2_gnu4.2.1/bin:/sw/xt/python/2.5.2/sles9.2_gnu3.3.3//bin:
/sw/xt/hdf5/1.6.7/cnl2.0_pgi7.1.6_par/bin:/proj/perc/TOOLS/tau_latest/x86_64/bin:./:
/ccs/home/rmills/bin:/spin/sys/adm/bin:/proj/perc/TOOLS/tau_latest/x86_64/bin:./:
/ccs/home/rmills/bin:/spin/sys/adm/bin:/opt/xt-lustre-ss/2.0.49a/usr/sbin:
/opt/xt-lustre-ss/2.0.49a/usr/bin:/opt/xt-boot/2.0.49a/bin/snos64:
/opt/xt-catamount/2.0.49a/bin/snos64:/opt/xt-os/2.0.49a/bin/snos64:
/opt/xt-service/2.0.49a/bin/snos64:/opt/xt-prgenv/2.0.49a/bin:/opt/xt-asyncpe/1.0c/bi
/opt/xt-pe/2.0.49a/bin/snos64:/opt/xt-pe/2.0.49a/cnos/linux/64/bin:
/opt/pgi/7.1.6/linux86-64/7.1/bin:/opt/MySQL/4.0.27/etc:
/opt/MySQL/4.0.27/libexec:/opt/MySQL/4.0.27/bin:/opt/moab-5.2.3/bin:
/opt/torque/default/bin:/sw/xt/bin:/opt/modules/3.1.6/bin:/ccs/home/rmills/bin:
```

```
/usr/local/bin:/usr/bin:/usr/X11R6/bin:/bin:/usr/games:/opt/bin:/opt/gnome/b
/opt/kde3/bin:/opt/xt-ofed-1.2.5.5/bin:/opt/xt-ofed-1.2.5.5/sbin:/opt/pathsc
/opt/bin:/opt/public/bin:/opt/bin:/opt/public/bin
MAIL=/var/mail/rmills
MODULE_VERSION=3.1.6
YOD_LOGFILE=syslog
CPU=x86_64
HDF5_CLIB=-I/sw/xt/hdf5/1.6.7/cnl2.0_pgi7.1.6_par/include
-L/sw/xt/hdf5/1.6.7/cnl2.0_pgi7.1.6_par/lib -lhdf5
-L/sw/xt/szip/2.1/sles9.2_pgi7.0.7/lib -lsz -lz

GNU_POST_COMPILE_OPTS= -B/opt/amd/binutils-070220-amd-sles9/bin
-B/opt/amd/binutils-070220-amd-sles9/bin
-march=barcelona -B/opt/amd/binutils-070220-amd-sles9/bin

ASYNCPE_VERSION=1.0c
PWD=/ccs/home/rmills/proj/pflotran_c65b3bc13590/src/pflotran
INPUTRC=/etc/inputrc
RCLOCAL_MYSQL=true
XTPE_COMPILE_TARGET=linux
TAU_THROTTLE=1
_LMFILES_=/opt/modulefiles/modules/3.1.6:/sw/xt/modulefiles/DefApps:
/opt/modulefiles/torque/2.2.0-snap.200707311754:/opt/modulefiles/moab/5.2.3
/opt/modulefiles/xt-binutils-quadcore/2.0.1:/opt/modulefiles/MySQL/4.0.27:
/opt/modulefiles/pgi/7.1.6:/opt/modulefiles/xt-libsci/10.2.1:
/opt/modulefiles/xt-mpt/3.0.0:/opt/modulefiles/xt-pe/2.0.49a:
/opt/modulefiles/xt-asyncpe/1.0c:/opt/modulefiles/PrgEnv-pgi/2.0.49a:
/opt/modulefiles/xt-service/2.0.49a:/opt/modulefiles/xt-libc/2.0.49a:
/opt/modulefiles/xt-os/2.0.49a:/opt/modulefiles/xt-catamount/2.0.49a:
/opt/modulefiles/xt-boot/2.0.49a:/opt/modulefiles/xt-lustre-ss/2.0.49a:
/opt/modulefiles/xtpe-target-cnl:/opt/modulefiles/Base-opts/2.0.49a:
/opt/xt-asyncpe/1.0c/modulefiles/xtpe-quadcore:
/sw/xt/modulefiles/hdf5/1.6.7_par:/sw/xt/modulefiles/python/2.5.2-alt:
/sw/xt/modulefiles/mercurial/1.0:/opt/modulefiles/xt-craypat/4.1.1

C_DIR=/opt/xt-libc/2.0.49a
EDITOR=vim
SYSTEM_USERDIR=/tmp/work/rmills
MODULEPATH=/opt/xt-asyncpe/1.0c/modulefiles:/opt/modulefiles:
/opt/modules/3.1.6:/sw/xt/modulefiles

TAU_MAKEFILE=/proj/perc/TOOLS/tau_latest/craycnl/lib/Makefile.tau-multipleco
LOADEDMODULES=modules/3.1.6:DefApps:torque/2.2.0-snap.200707311754:
moab/5.2.3:xt-binutils-quadcore/2.0.1:MySQL/4.0.27:pgi/7.1.6:xt-libsci/10.2
xt-mpt/3.0.0:xt-pe/2.0.49a:xt-asyncpe/1.0c:PrgEnv-pgi/2.0.49a:xt-service/2.0
xt-libc/2.0.49a:xt-os/2.0.49a:xt-catamount/2.0.49a:xt-boot/2.0.49a:
xt-lustre-ss/2.0.49a:xtpe-target-cnl:Base-opts/2.0.49a:xtpe-quadcore:
hdf5/1.6.7_par:python/2.5.2-alt:mercurial/1.0:xt-craypat/4.1.1

PS1=$HOSTNAME:${PWD#$HOME/}>
```

```
PGI_POST_COMPILE_OPTS= -Ya,/opt/amd/binutils-070220-amd-sles9/bin
-tp barcelona-64 -Ya,/opt/amd/binutils-070220-amd-sles9/bin
-Ya,/opt/amd/binutils-070220-amd-sles9/bin

LM_LICENSE_FILE=/opt/pgi/7.1.6/license.dat
XTPE_QUADCORE_ENABLED=ON
TEXINPUTS=:/ccs/home/rmills/.TeX:/usr/share/doc/.TeX:/usr/doc/.TeX
HOME=/ccs/home/rmills
SHLVL=2
CRAYPAT_PRE_COMPILE_OPTS= ‘$PAT_ROOT/sbin/pat-opts PRE_COMPILE‘
OSTYPE=linux
LESS_ADVANCED_PREPROCESSOR=no
PGI_PATH=/opt/pgi/7.1.6
PTL_SNOS_NAL=SS
PAT_RT_EXPERIMENT=samp_cs_time
CRAYPAT_INCLUDE_OPTS= ‘$PAT_ROOT/sbin/pat-opts INCLUDE‘
PAT_ROOT=/opt/xt-tools/craypat/4.1.1/cpatx
XCURSOR_THEME=
LS_OPTIONS=-N --color=tty -T 0
LIBLUSTRE_DEBUG_CONSOLE=0
SE_DIR=/opt/xt-service/2.0.49a
GTK_PATH=/usr/local/lib64/gtk-2.0:/opt/gnome/lib64/gtk-2.0:/usr/lib64/gtk-2.0
LOGNAME=rmills
MACHTYPE=x86_64-suse-linux
LESS=-M -I
PYTHONPATH=/sw/xt/mercurial/1.0/sles9.2_gnu4.2.1/lib/python2.5/site-packages
CVS_RSH=ssh
SSH_CONNECTION=128.219.176.51 50923 160.91.205.218 22
TAU_OPTIONS=-optPreProcess -optVerbose
CRAYPAT_POST_LINK_OPTS= ‘$PAT_ROOT/sbin/pat-opts POST_LINK‘
MODULESHOME=/opt/modules/3.1.6
CRAYPAT_PRE_LINK_OPTS= ‘$PAT_ROOT/sbin/pat-opts PRE_LINK‘
LESSOPEN=lessopen.sh %s
LIBSCI_BASE_DIR=/opt/xt-libsci/10.2.1
INFOPATH=/opt/MySQL/4.0.27/info:/usr/local/info:/usr/share/info:/usr/info
LIBSCI_VERSION=10.2.1
HDF5_FLIB=-module . -module /sw/xt/hdf5/1.6.7/cnl2.0_pgi7.1.6_par/lib
-I/sw/xt/hdf5/1.6.7/cnl2.0_pgi7.1.6_par/include
-L/sw/xt/hdf5/1.6.7/cnl2.0_pgi7.1.6_par/lib -lhdf5_fortran
-lhdf5 -L/sw/xt/szip/2.1/sles9.2_pgi7.0.7/lib -lsz -lz

DISPLAY=localhost:12.0
SVNROOT=file:///spin/home/rmills/svnrepos
XAUTHLOCALHOSTNAME=jaguar10
LESSCLOSE=lessclose.sh %s %s
PE_DIR=/opt/xt-pe/2.0.49a
LIBSCI_POST_LINK_OPTS= -lsci_quadcore
PETSC_DIR=/ccs/proj/geo002/petsc-dev
G_BROKEN_FILENAMES=1
COLORTERM=1
LUSTRE_DIR=/opt/xt-lustre-ss/2.0.49a
```

```
_=/usr/bin/env
```

0.24.2 Compilation of PFLOTRAN on `jaguar.ccs.ornl.gov`

```
jaguar10:proj/pflotran_c65b3bc13590/src/pflotran> hg tip
changeset:   1281:c65b3bc13590
tag:         tip
parent:      1280:acd7ce3bf564
parent:      1279:683312f7276d
user:        Glenn Hammond
date:        Thu Jul 03 15:54:18 2008 -0700
summary:     merge

jaguar10:proj/pflotran_c65b3bc13590/src/pflotran> pushd $PETSC_DIR
/ccs/proj/geo002/petsc-dev ~/proj/pflotran_c65b3bc13590/src/pflotran
jaguar10:/ccs/proj/geo002/petsc-dev> hg tip
changeset:   12336:fc1707c94fca
tag:         tip
user:        hzhang@localhost.localdomain
date:        Tue Jul 01 05:25:49 2008 -0500
summary:     MatGetInertia() should not be used when PETSC_USE_COMPLEX

jaguar10:/ccs/proj/geo002/petsc-dev> popd
~/proj/pflotran_c65b3bc13590/src/pflotran

jaguar10:proj/pflotran_c65b3bc13590/src/pflotran> make clean
jaguar10:proj/pflotran_c65b3bc13590/src/pflotran> make hdf5=1 jaguar=1 pflot

ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev
-I/ccs/proj/geo002/petsc-dev/cray-xt4-pgi_craypat/include
-I/ccs/proj/geo002/petsc-dev/include
-I/autofs/na2_proj/geo002/petsc-dev/cray-xt4-pgi_craypat/include
-I/autofs/na2_proj/geo002/petsc-dev/cray-xt4-pgi_craypat/include
-I/opt/mpt/3.0.0/xt/mpich2-pgi/include
-I./cray-xt4-pgi_craypat/obj -module .
-module /sw/xt/hdf5/1.6.7/cnl2.0_pgi7.1.6_par/lib
-I/sw/xt/hdf5/1.6.7/cnl2.0_pgi7.1.6_par/include
-L/sw/xt/hdf5/1.6.7/cnl2.0_pgi7.1.6_par/lib -lhdf5_fortran
-lhdf5 -L/sw/xt/szip/2.1/sles9.2_pgi7.0.7/lib -lsz -lz
-DUSE_HDF5  -o fileio.o fileio.F90

/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used

fileio.F90:

ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev
-I/ccs/proj/geo002/petsc-dev/cray-xt4-pgi_craypat/include
-I/ccs/proj/geo002/petsc-dev/include
-I/autofs/na2_proj/geo002/petsc-dev/cray-xt4-pgi_craypat/include
```

```
-I/autofs/na2_proj/geo002/petsc-dev/cray-xt4-pgi_craypat/include
-I/opt/mpt/3.0.0/xt/mpich2-pgi/include
-I./cray-xt4-pgi_craypat/obj -module .
-module /sw/xt/hdf5/1.6.7/cnl2.0_pgi7.1.6_par/lib
-I/sw/xt/hdf5/1.6.7/cnl2.0_pgi7.1.6_par/include
-L/sw/xt/hdf5/1.6.7/cnl2.0_pgi7.1.6_par/lib -lhdf5_fortran
-lhdf5 -L/sw/xt/szip/2.1/sles9.2_pgi7.0.7/lib -lsz -lz
-DUSE_HDF5  -o water_eos.o water_eos.F90

/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used

water_eos.F90:

ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p

/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used

gaseos_mod.F90:

ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p

/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used

co2eos.F90:

ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p

/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used

co2_sw_rtsafe.F90:

ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p

/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used

co2_span_wagner.F90:

ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p

/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used

spline.F90:

ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p

/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used

co2_span_wagner_spline.F90:

ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
```

```
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
co2_sw.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
utility.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
option.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
logging.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
reaction_aux.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
reactive_transport_aux.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
region.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
field.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
material.F90:
```

```
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
richards_aux.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
richards_lite_aux.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
pckr_mod.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
mphase_aux.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
auxilliary.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
debug.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
waypoint.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
strata.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
```

```
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
connection.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
breakthrough.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
solver.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
cf90bridge.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
unstructured_grid.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
structured_grid.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
condition.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
grid.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
coupler.F90:
```

```
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
hydrostatic.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
patch.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
level.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
amrgrid.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
discretization.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
reaction.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
realization.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
hdf5.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
```

```
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
mphase.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
transport.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
reactive_transport.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
richards.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
richards_lite.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
mass_balance.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
general_grid.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
checkpoint.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
convergence.F90:
```

```
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
output.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
timestepper.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
simulation.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
init.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
amrstubs.F90:
ftn -c  -tp barcelona-64 -fastsse   -I/ccs/proj/geo002/petsc-dev -I/ccs/proj/geo002/p
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
pflotran.F90:
ftn -tp barcelona-64 -fastsse     -o pflotran ./fileio.o ./water_eos.o ./gaseos_mod.o
/opt/xt-asyncpe/1.0c/bin/ftn: INFO: linux target is being used
File with unknown suffix passed to linker: /opt/pgi/7.1.6/linux86-64/7.1-6/lib/pgi.lo
File with unknown suffix passed to linker: /opt/pgi/7.1.6/linux86-64/7.1-6/lib/pgi.lo
File with unknown suffix passed to linker: /opt/pgi/7.1.6/linux86-64/7.1-6/lib/pgi.lo
/ccs/proj/geo002/petsc-dev/cray-xt4-pgi_craypat/lib/libpetsc.a(ghome.o)(.text+0x23)/u
: In function ‘PetscGetHomeDirectory’:
```

```
: warning: Using 'getpwuid' in statically linked applications requires at ru
/ccs/proj/geo002/petsc-dev/cray-xt4-pgi_craypat/lib/libpetsc.a(send.o)(.text
: In function 'SOCKCall_Private':
: warning: Using 'gethostbyname' in statically linked applications requires
jaguar10:proj/pflotran_c65b3bc13590/src/pflotran> hg ls -l pflotran
m-rwxr-xr-x  1 rmills ccsstaff 60765580 Sep 23 23:13 mpflotran
jaguar10:proj/pflotran_c65b3bc13590/src/pflotran> exit
```

0.24.3 Running PFLOTRAN on `jaguar.ccs.ornl.gov`

To execute the binary, one submits the job with the following run script.

```
#!/usr/bin/csh
#PBS -A CSC044
#PBS -N pflotran_300A_hwpc_128
#PBS -j oe
#PBS -V
#PBS -l walltime=2:00:00,feature=800,size=4000

cd $PBS_O_WORKDIR
date
nodeinfo

. /opt/modules/default/etc/modules.sh
module load xt-craypat/4.1.1
cd $PBS_O_WORKDIR
date
nodeinfo

pat_hwpc aprun -n $PBS_NNODES ./pflotran -snes_ls basic -log_summary
```